

Poznan University of Technology  
Faculty of Mechanical Engineering  
Division of Virtual Engineering

A thesis submitted for the fulfilment of the degree of Doctor of Philosophy

**BUILDING PARAMETRIC GEOMETRIC MODELS FOR  
CAD SYSTEMS BASED ON TOPOLOGICAL  
OPTIMIZATION RESULTS**

Jan Polak

Supervisor  
prof. Michał Nowak

Poznań, 2023

# Contents

<b>Contents</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
<b>2 Background</b>	<b>5</b>
<b>3 Biomimetic structural optimization approach</b>	<b>8</b>
<b>4 Research aim and objectives</b>	<b>15</b>
<b>5 Biomimetic structural optimization system</b>	<b>17</b>
5.1 Evaluation of the mesh generator tool . . . . .	17
5.2 Recommended changes to the mesh generator tool . . . . .	19
<b>6 3D Shapes Recognition using Hough Transform</b>	<b>23</b>
6.1 Orientation Estimation . . . . .	24
6.2 Position and Radius Estimation . . . . .	26
6.3 Shape Ends Positions Estimation . . . . .	28
6.4 Post-recognition step . . . . .	28
6.5 Implementation notes . . . . .	29
6.6 Algorithm Validation Methodology . . . . .	29
<b>7 Automated, Skeletonization-Based Truss Recognition</b>	<b>31</b>
7.1 Preprocessing . . . . .	33
7.2 Skeletonization . . . . .	33
7.3 Skeleton Processing . . . . .	34
7.4 Heuristic Refinement . . . . .	36
7.5 Postprocessing and CAD Integration . . . . .	38
7.6 Solution Verification—Methodology . . . . .	39
<b>8 Application of the parametric model in the hybrid approach</b>	<b>41</b>
<b>9 Results</b>	<b>45</b>
9.1 Changes in the Mesh Generator Tool . . . . .	46
9.2 3D Shape Recognition using Hough Transform . . . . .	49
9.2.1 Case Study 1—Single Capped Cylinder . . . . .	49

9.2.2	Case Study 2—Complex Parametric Object . . . . .	53
9.2.3	Case Study 3—Optimized Cantilever Beams . . . . .	57
9.3	Automated, Skeletonization-Based Truss Recognition . . . . .	65
9.3.1	Comparison with the Literature . . . . .	65
9.3.2	Case Study 1—Cantilever Bar Bending . . . . .	66
9.3.3	Case Study 2—Box Corner . . . . .	70
9.4	Application of the proposed workflow to the industrial use case. . . . .	79
<b>10</b>	<b>Discussion</b>	<b>85</b>
10.1	Mesh generator tool . . . . .	85
10.2	Hough Transform . . . . .	86
10.3	Skeletonization-based approach . . . . .	87
10.4	Industrialization with the hybrid approach . . . . .	88
<b>11</b>	<b>Conclusions and future work plan</b>	<b>90</b>
11.1	Conclusions . . . . .	90
11.2	Future Work . . . . .	91
	<b>List of Figures</b>	<b>93</b>
	<b>List of Tables</b>	<b>101</b>
	<b>Nomenclature</b>	<b>102</b>
	<b>Bibliography</b>	<b>103</b>

# Chapter 1

## Introduction

Design of lightweight structures, which have both low mass and good mechanical properties, became a crucial part of the mechanical design process. It is achieved not only by using advanced lightweight materials, but mainly by using the advanced design process. Leading industrial approach is the use of structural optimization to design both the stiffest and lightweight design. Combined with Additive Manufacturing techniques [1–3], structural optimization is an excellent method for achieving solutions greatly exceeding what is possible to create using traditional techniques of design and manufacturing.

Structural optimization software, by design, is based on the structural analysis numerical methods. Those methods, with Finite Elements Method as the dominant approach, require discretization of the analyzed area. This leads to discretized results of structural optimization - surface or volumetric finite element mesh. Additive manufacturing production methods may reduce post-processing of structural optimization results only to the mesh surface smoothing. In many practical cases, in particular when after optimization further design requires a parametric model, there is a need of transferring structural optimization results to the CAD system. This requires building a parametric model.

Although there are multiple structural optimization methodologies, the challenge of transition from finite element mesh to parameterized model is always the same. Depending on the selected manufacturing method it is possible to limit optimization results processing to surface smoothing and continue the design process by relying on the manufacturing method preparation steps. Separation of the design optimization step and the manufacturing preparation step simplifies overall manufacturing workflow but in practice is often insufficient - it is impossible to backtrack from the latter step and introduce adjustments other than direct operations on a mesh. Conversion of structural optimization results back into parametric models is an important research area that is vital not only for preparation for the manufacturing process, but also when the mechanical design process requires, after the structural optimization step, more design cycles based on parametric models in CAD systems.

The purpose of this thesis is the research of building parametric geometric models for CAD systems based on topological optimization results. Research is conducted in two areas. First is research of available feature recognition techniques and application in the area of parametric model building. Second is application of the parametric models in industrial

application. The research aims to incorporate developed parametric model recognition algorithms into a consistent workflow to bridge the gap between the structural optimization process and the manufacturing in the mechanical design process.

## Chapter 2

# Background

Structural optimization is used as one of many steps in the Additive Manufacturing design workflow which starts with product planning and ends at product validation [4]. Optimized part is further processed in CAD software to post-process it and prepare for manufacturing [5]. Hence it is useful to have structural optimization output in a format that allows easy modification and processing. The CAD parametric model meets those requirements. In practice conversion to the parametric model requires a trade off between the model accuracy and the ease of processing; different techniques approach this in different ways. Tracing models using parametric surfaces can very accurately reflect structural optimization complexity but manual processing or adjusting models can be as difficult as processing the mesh itself. On the other hand, feature extraction approaches allow conversion of the structural optimization output into a set of simple features that are easy to modify but may lead to oversimplification of the result. The right approach must be taken depending on the use case and requirements of the design process.

Features extraction in the 2D domain is well explored. Structural optimization output, especially obtained with the SIMP method, can be treated like a black-and-white picture so a classical computer vision algorithm can be applied. Simple features such as circles, lines can be extracted using the Hough transform [6]. Depending on the use case it is possible to use features mapping to map topology of the optimization output [7]. In this approach complexity of the model is reduced into a topology graph which can be converted in the CAD software into truss. Opposite approach is the use of curve fitting to create a parametric model of the optimized object surface [8]. This approach doesn't provide any knowledge about the optimized object directly but accurately maps the boundary of the object in 2D space as a parametric model. Morphological image processing is another technique from the domain of computer vision adapted for the structural optimization results processing [9]. Skeletonization capabilities provided by the morphological processing enables feature extraction such as partitioning of the input data or extracting connectivity graph. The challenge in such approaches is filtering of the skeleton. Classical morphological filtering is sensitive to input data and additional processing is required, either of the input data or of the resulting skeleton.

Compared to well-explored 2D approaches, selection of 3D techniques is limited. Greatly increased topological complexity of 3D models is the main limiting factor [10]. Moreover

computer vision approaches cannot be applied to 3D shapes limiting applying achievements from this board branch of research to the area of mechanical design.

Structural optimization in the 3D domain creates very complex topologies and shapes, which are difficult to represent as parametric models in CAD systems. Multiple approaches completely ignore this problem and do not use parametric models to prepare models for manufacturing. For additive manufacturing approaches using layer by layer approaches such as Selective Laser Sintering (SLS) or Powder Bed Fusion (PBF) there is no need for volumetric mesh as slicer software requires surface shell used to create codes for the manufacturing device [11]. Hence it is enough that the mesh is watertight shell free of defects [12]. A water-tight shell from the structural optimization system is often not enough because the optimization procedure may produce rough surfaces, with small defects or peaks. The leading approach to improve mesh quality is surface smoothing [13–18]. Surface smoothing techniques change position of mesh vertices using Laplace operator, local curvature of neighbor faces, filter based on patch normal or filtering surface with a frequency based approach. Each method has different advantages and disadvantages but volume shrinkage is shared by virtually all approaches. This phenomenon is especially dangerous in mechanical engineering, where lightweight structures are designed to carry loads within designed range. Volume reduction changes mechanical properties of the design and may lead to failure. Another problem is distortion of flat surfaces, which become curved with increased level of smoothing. This causes problems when the mesh contains structural features excluded from optimization such as mount points. Smoothing distorts those mesh areas and dedicated processing is required to restore original shapes.

Separate branch of research in the parametric model creation area is the family of 2.5D techniques. Structural optimization preparation for manufacturing using classical approach – chip removal techniques – is done by preparing a set of material removal instructions starting from a solid block of metal. Parametric models prepared this way can be easily, by the virtue of using the same commands, converted into a program for CNC machines. Algorithms working this way have access only to the surface of the model, hence the name – 2.5D. This limitation greatly decreases numeric complexity required to find the solution. Available techniques create parametric solution by matching pre-defined 2D shape templates [19] or by matching machining features using dedicated algorithms [20] or machine learning [21,22]. All those approaches require defining a set of input features to be matched against optimization results. In practice this is not a limiting factor as those features are derived from the set of machining tools available on the manufacturing machine.

Full parametric model of a 3D object can be created using cubic B-splines. This approach allows tracing surface curvature using a mathematics model. Until recently semi-automated approaches were available. Algorithm creates a skeleton which is used as a leading curve that guides the surface created in the CAD software by the operator [23]. Computer-aided, semi-automatic design speeds up the creation process of the parametric model in approaches where a parametric model is required for the manufacturing process. There is no need to create the model from scratch manually, based on the structural optimization output which results in time savings and higher quality of the result model.

Fully automated approach was recently presented [24]. In this approach a skeleton is used to find mesh junctions, which are converted into parametric representation using B-splines. Junctions are connected together using lofting curves. Created model is fully parametric, smooth and watertight. Both presented approaches rely on the skeletonization algorithm to discover the initial topology, which is used as the starting point for the feature recognition step. Extra care must be taken to correctly process the skeleton as in practical cases the generated skeleton contains errors such as loose ends, not needed branches or is not centrally located inside the mesh which causes problems in subsequent steps.

## Chapter 3

# Biomimetic structural optimization approach

This chapter describes the biomimetic structural optimization system used in the research. The system uses an approach based on trabecular bone remodelling phenomenon. Similarities between bone remodelling and structural optimization are described and details of the system are provided.

The trabecular bone structure is continually rebuilt. This process is the subject of intensive research due to its high medical significance [25–31]. The observation proposed by Julius Wolff - called the Wolff's law [32] - can be described as a structural adaptation of the bone to the external forces. A model of the trabecular bone remodeling phenomenon is based on the idea of a regulatory model presented by Huiskes [33–35]. The lazy zone concept - proposed by Carter [36] - is also included in the presented model, but this issue will be discussed in a section devoted to the simulation approach. The phenomenon of trabecular bone adaptation has two important attributes. First, mechanical stimulation is needed to conserve the rebuilding balance. Second, the process of resorption and formation occurs only on the trabecular bone surface. The process takes place within Basic Multicellular Unit (BMU), areas with a smaller volume, but comparable to a small part of a single trabecula. This is where the sequence of resorption and new tissue formation takes place. Adding the ability to determine the level of local mechanical stimulation, all the actors needed to build the model are now present. The model consists of a regulatory mechanism (on the bone surface only) between bone resorption and formation, corresponding to the intensity of mechanical stimulation and this is the main assumption of the discussed model. The regulatory mechanism depends, in turn, on the mechanical stimulation of the entire bone structure. The assumption based on clinical observations is that, if the strain energy density is close to observed equilibrium, there is no change in the bone mass during the bone remodeling process. In this way, if the intensity of mechanical stimulation differs slightly from the homeostatic value of the strain energy density, bone mass does not change. However when the intensity of mechanical stimulation is larger or smaller than the reference value, the bone mass increases or decreases respectively. Modeling the actual bone geometry allows the simulation of the bone remodeling phenomenon in the most similar way to the process occurring on the surface of the BMU's. Hence the necessity of geometrical modeling

of the real geometry of the reconstructed structure becomes clear.

Continuous models, although they currently dominate the area of modeling the evolution of the trabecular bone structure, are not sufficient to simulate real processes on the surface of the bone structure [37–39]. The microarchitecture of bone tissue has a major influence on the mechanical properties and must also be considered when trying to explain disease processes.

In order to be able to provide general relationships between the geometric form of the structure observed in imaging studies and its mechanical properties, an accurate, three-dimensional simulation model of the trabecular bone structure is necessary. As indicated above, there are two elements necessary to build a computing environment - the inclusion of multiple load cases and an accurate geometric model of the trabecular tissue. The approach used by the structural optimization system uses two postulates [40] concerning the trabecular bone remodeling regulatory model, based on shape optimization studies [41, 42]. These studies show that the remodeling of the trabecular bone can be treated as a simultaneous optimization of shape and topology. The similarity between the phenomenon of trabecular bone remodeling and topology optimization has been recognized and it is used in two opposite research approaches. The first approach is to use exact mathematical results from the optimization area (maximization or minimization of a function) to predict the evolution of the bone structure [43–45]. And the second approach which is presented also in the paper is to apply the bio-mechanical observations and models to the structural optimization issues [41, 46, 47]. In the latter paper it was proved with use of shape derivative, that the maximization of a structure stiffness needs the structural form, having on the part of the boundary, subject to modification, constant value of the strain energy density. This is also the purpose of the trabecular bone remodeling phenomenon.

The postulates are as follows:

**Postulate 1.:** during the remodeling process, the trabecular bone tends to maximize the stiffness of a structure (i.e. to find the stiffest design) by the strain energy density equalization on the structural surface of the trabecular tissue. According to formula:

$$\sigma(\mathbf{u}) : \varepsilon(\mathbf{u}) = \lambda = \text{const.} \quad (3.1)$$

where:

$\sigma(\mathbf{u}) : \varepsilon(\mathbf{u})$  - strain energy density at the point on structural surface

$\lambda$  - homeostatic value of strain energy density (surrounded by the lazy zone)

the bone remodeling phenomenon can be interpreted as a structural optimization process. It means that for the stiffest design, the strain energy density on the part of the boundary subject to modification must be constant. Comparing this result with the regulatory model of bone remodeling, it can be deduced that the mechanical stimulation (measured by strain energy density distribution on structural surface) has to be between the values determined by the lazy zone. In other words, at each point of the surface of the trabecular bone, the strain energy density should be similar, and this can be achieved by remodeling the bone structure.

**Postulate 2.:** the regulatory model of the trabecular bone remodeling can be applied to the multiple load problem maintaining its character and the basic assumptions about the existence of the homeostatic value of the strain energy density. The 'lazy zone' is an important element of the model which provides the opportunity to find a solution for many load cases. The local change (in this case related to different loads) on the structural surface leads to global minimization of the strain energy for the whole structure.

According to the formula:

$$\alpha_1\sigma(\mathbf{u}_1) : \varepsilon(\mathbf{u}_1) + \alpha_2\sigma(\mathbf{u}_2) : \varepsilon(\mathbf{u}_2) = \lambda = \text{const.} \quad (3.2)$$

$$\alpha_1\sigma(\mathbf{u}_1) : \varepsilon(\mathbf{u}_1) + \alpha_2\sigma(\mathbf{u}_2) : \varepsilon(\mathbf{u}_2) = \lambda = \text{const.} \quad (3.3)$$

where:

$\alpha_1\sigma(\mathbf{u}_1) : \varepsilon(\mathbf{u}_1) + \alpha_2\sigma(\mathbf{u}_2) : \varepsilon(\mathbf{u}_2)$  - weighted sum of strain energy density at the point on structural surface for different load cases

$\lambda$  - homeostatic value of strain energy density (surrounded by the lazy zone)

the regulatory model can be directly used for the simulation of the multiple load problem.

To carry out the biomimetic structural optimization simulation based on the trabecular bone remodelling phenomenon it is necessary to combine two areas - numerical simulation of deformation of the structure under load and structural evolution. The Finite Element Method will be used for this purpose. In terms of structural calculations, there is no alternative but to use the already existing FEM systems. Described structural optimization system is able to use FrontISTR [48], Elmer [49] and Calculix [50] for this purpose. The more difficult task is to plan how to reflect the evolution of the structure. Based on previous experience [51], it was decided to build separate computational meshes for each simulation step. The idea of building a volumetric mesh based on data in the form of two-dimensional images containing the cross-sections of tissue is presented schematically in Figure 3.1. This is a natural way of mesh generation since the visualization for the biological entities is based on the digital images e.g. Computer Tomography. Also the evolution of the structure is based on the two-dimensional images modification, which is depicted schematically in Figure 3.2. According to the remodeling scenario described by the regulatory model, depending on the calculated value of the energy density, the surface of the structure is modified by adding or removing material on its surface.

Input of the system is data set exported from the CAD system as series of STL files. Definition of the optimization task consists of:

- Main mesh to be optimized,
- Boundary areas – as optimization progresses shape and topology changes. Pre-defined boundary areas define where support and forces are applied each step,
- Domain area – bounding box defining the area of operation for the system,

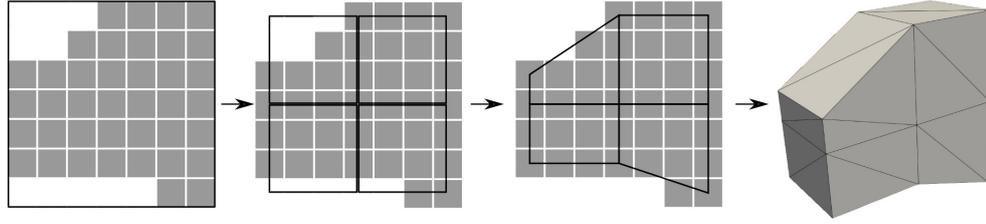


FIGURE 3.1. Data Discretization and Volumetric Mesh Building algorithms applied for sample image. Image is divided into squares, in this particular example size of 3 was selected. For each square, four points are selected and used for volumetric mesh building. Last image shows a volumetric mesh created from two identical images.

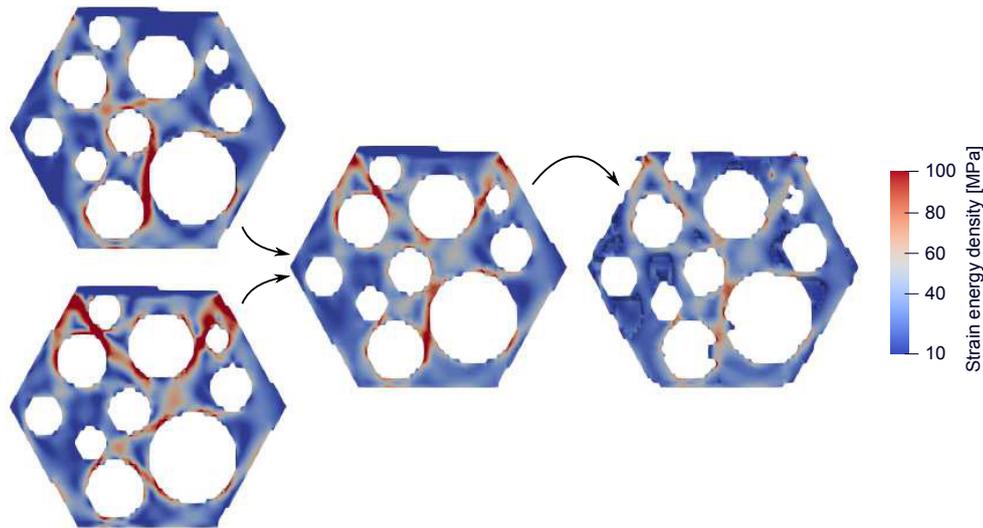


FIGURE 3.2. Showcase of the multiple load case and mesh evolution algorithm. On the left element with random holes subjected to shear stress (top) and compression (bottom). In the middle both cases are aggregated using formula 3.2. On the right the same element after the evolution algorithm has been applied and then both load cases are calculated and aggregated together. New material 'grows' in high and is removed from low strain energy density areas. Material with strain energy density values inside the lazy zone is left intact.

- Areas excluded from the optimization – those parts won't be modified during the remodelling step. If initially there is material in those areas it will be retained through the optimization,
- Multiple load scenario definition – file which defines, for each multiple load scenario, how forces are connected to defined boundary areas.

Sample problem definition, as seen in CAD is shown in Figure 3.3 and multiple load scenario definition in Figure 3.4.

System created has been designed with two main goals: high performance and flexibility for future enhancements and modifications. Underlying phenomenon of trabecular bone remodeling is a process that happens over time hence the algorithm works in iterations, each run is a full remodeling cycle. Single loop consists of a series of steps. First step is data discretization to reduce the size of data used to build mesh. Scale factor is input parameter

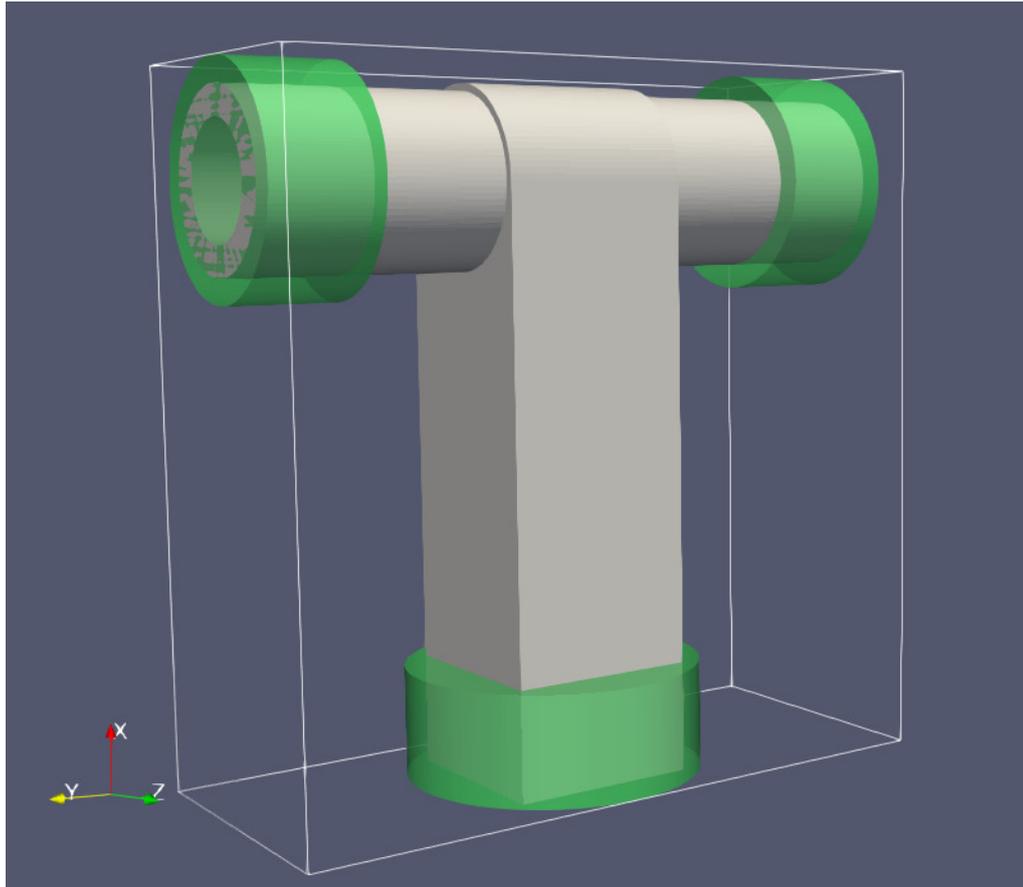


FIGURE 3.3. Sample input for the structural optimization system, as seen in a CAD system. Gray – the main element. Green – three distinct boundary areas. White outline – optimization domain bounding box.

for algorithm and allows choosing between fine and coarse meshes and consequent trade-off between solution resolution and resources required to perform calculations. Each two-dimensional image (slice) is divided into squares of size depending on scale factor that are used as an input for Volumetric Mesh Building. For each square up to 4 points on edges are selected in such a way that the resulting tetragon approximates material inside each square. Second step is the Volumetric Mesh Building and it consists of three operations. First, pairs of adjacent discretized images are converted into layers of mesh cells – squares from adjacent two-dimensional images are paired and for each pair, mesh cells are created, each consisting of 6 tetrahedral elements spanned between points from adjacent squares. Data is filtered to remove mesh items that are not eligible for strain energy density (SED) calculation, mesh items that are connected only by a single point or single edge. Filtered data is collected and merged into a single mesh – each cell and vertex gets assigned a number and boundary faces are computed. Prepared mesh is saved and FEM software is used to calculate displacement and SED values. To support multiple load scenarios, for each load case, separate simulation is executed in a loop. Results of all calculations are merged together according to equation 3.2. Provided formula is not limited to two cases and thus implementation supports any amount of load cases, each allowing a different set of forces and boundary conditions. The only limiting factor is time - each load case

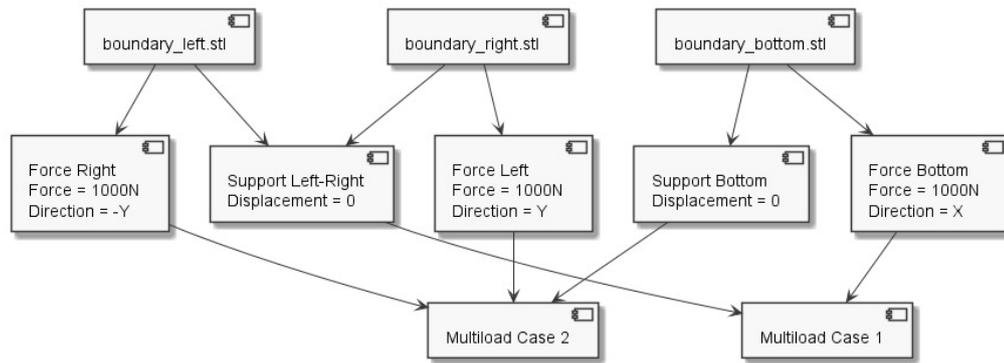


FIGURE 3.4. Sample visualization of multiple load scenario definition. Defined boundary areas can have assigned different boundary conditions in each load case scenario allowing flexible optimization task definition. In the shown example *boundary\_left.stl* is supported in the *Multiload Case1* and stressed area in *Multiload Case2*.

requires a separate SED calculation which is the most time consuming part of the whole process. After all load cases are evaluated, the stop criterion is evaluated - if surface SED values are all inside the lazy zone or maximum allowed number of iteration is exceeded, algorithm stops. SED data is passed to the Mesh Evolution algorithm where it is applied onto two-dimensional images used for mesh building. Each image is remodelled depending on a surface SED – areas where SED exceeds the upper bound of the lazy zone grow and those where SED is below the lower bound shrink. Remodeled stack of images is input for the next iteration. Each iteration of the algorithm generates a valid solution, refined with each step. Sample output is shown in Figure 3.5.

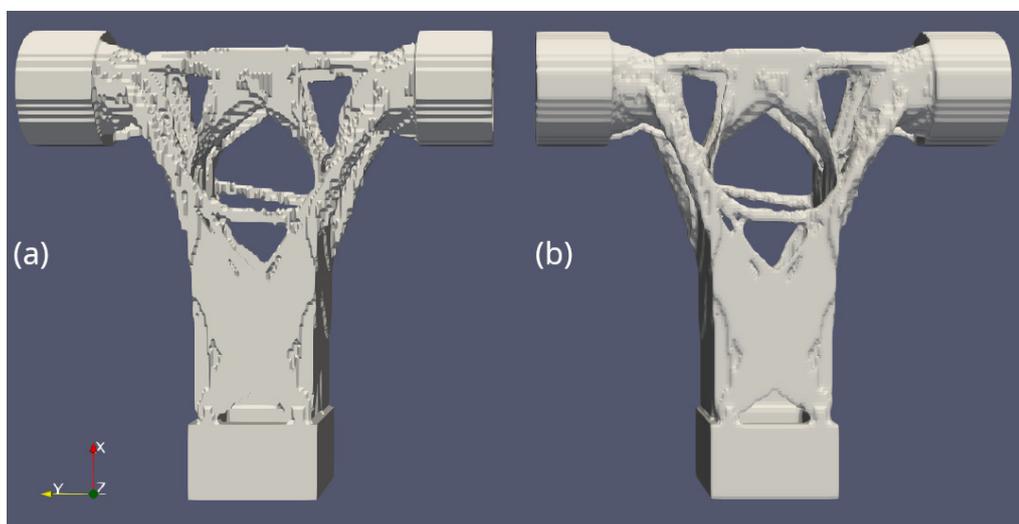


FIGURE 3.5. Sample output mesh of the biomimetic structural optimization system, obtained for the input data from Figure 3.3. (a) Optimized mesh. (b) Optimized mesh with smoothing applied, for visualization purposes only.

Presented structural optimization system was selected for two main reasons. First one is ability to operate on the direct mesh. This is the distinctive feature of this approach, which is not present in the leading structural optimization approach – Solid Isotropic Material with Penalization (SIMP). The SIMP method produces a density map which requires an

additional post-processing step to obtain the proper solution consisting of finite element mesh. Presented biomimetic approach operates on the mesh directly, so no additional step is required. Proposed feature recognition framework will be able to operate directly on the optimization output. The second reason is related to the first one - no need to process optimization is possible because the optimization system contains its own mesh generator. It is possible to reuse generator during research work.

## Chapter 4

# Research aim and objectives

The research aims to provide the missing step between the structural optimization system and manufacturing process by introducing the parametric model of the structural optimization system results. The ability to use such a model enables preparing a solution for the manufacturing step by including technological constraints in the model. Two observations are the entry point for the research. First is the observation that output of the biomimetic structural optimization contains a broad class of solutions resembling truss structure with multiple bar-like elements. Second observation comes from the author's own experience. The main demand from the manufacturing team during the model preparation is to increase the thickness of the model's small elements. The first observation directs the research in the direction of truss recognition systems. Second suggests choosing a parametric model that allows easy modification of parts during the manufacturing preparation process. This led to focus on algorithms allowing an automated recognition of the truss structure that can be converted into a parametric CAD model. To achieve this several objectives are identified:

- Assess the current state of the mesh generator in the structural optimization system. High quality of the input mesh is considered as an important factor in the features recognition system. The mesh of high quality has smooth surface and free of topology defects. Identified problems must be addressed before further research.
- Provide Hough Transform implementation for detecting cylinders in the structural optimization results. Hough Transform is an algorithm commonly used for feature detection in computer vision. High accuracy, fast operations and successful application in the computer vision area make it an attractive target for the research.
- Explore skeletonization-based approach for features recognition. Mesh skeletonization algorithms provide a convenient way of detecting the topology, which can be used to rebuild objects using simple features. For a complete approach additional techniques for skeleton filtering and refinement have to be proposed.
- Propose steps to incorporate research results in the end-to-end manufacturing process. Use industrial processes exercised in the BioniAMoto project to demonstrate usefulness of the parametric model in the industrial application.

## Thesis outline

The remainder of the thesis is structured as follows:

- Chapter 5 assesses the usability of the biomimetic structural optimization mesh generator system for industrial use. Available tools are used to evaluate quality of result mesh and changes to the system are proposed.
- Chapter 6 presents the application of the Hough Transform algorithm for cylinders detection in the area of parametric model reconstruction. The algorithm presented was tailored for the use with finite element meshes.
- Chapter 7 introduces skeletonization-based approach for automated truss recognition. All steps of the approach are broadly presented with examples.
- Chapter 8 explores potential applications of research in the mechanical design process. Based on the prior experience the hybrid approach is proposed. In the described workflow, the researched parametric model, after manufacturing process limitations are addressed, is fused back into the smoothed source mesh.
- Chapter 9 presents results of research. In distinct sections each of previous areas of research – mesh generator evaluation, cylinders detection based on the Hough Transform, skeletonization-based truss recognition and hybrid approach – are evaluated using test data containing both synthetic and real test cases.
- Chapter 10 discusses the proposed developments and provides a critical overview of results of the research.
- Chapter 11 presents conclusion of the research, highlights opportunities and proposes future directions for the developments.

## Chapter 5

# Biomimetic structural optimization system

In this chapter the mesh generator included in the biomimetic structural optimization system is evaluated. The quality of the mesh is assessed using a mesh analysis tool available in the FreeCAD software. Issues are identified and algorithm solving issues are provided.

### 5.1 Evaluation of the mesh generator tool

The mesh generator works using a slice by slice approach. Two neighboring data layers are analyzed at once and cubic structural elements are spanned across lattice points, as described in detail in the Chapter 3 and shown in a Figure 5.1. Each cubic element consists of 6 tetrahedral elements. To assess the quality of the generator mesh was generated using

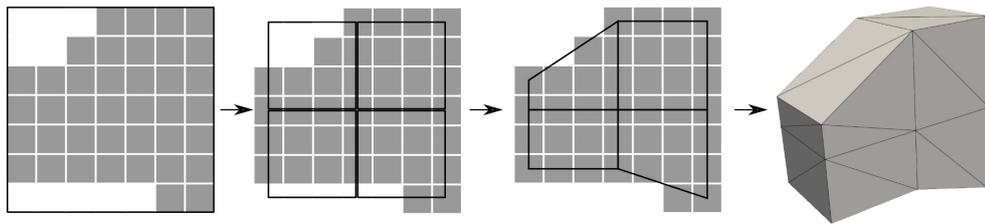


FIGURE 5.1. Data Discretization and Volumetric Mesh Building algorithms applied for a sample image. The image is divided into squares, in this particular example size of 3 was selected. For each square, four points are selected and used for the volumetric mesh building. The last image shows volumetric mesh created from two identical images.

the micro-CT of rat femur bone (Figure 5.2). Resulting mesh is big and very complex (STL file has 104MB, contains 2131028 cells and 1062076 points). Detail of the mesh is shown in Figure 5.3. Mesh was analyzed using the FreeCAD mesh evaluator. This tool is able to detect following issues in a surface mesh:

- Each triangular face in the mesh must have, by convention, points defined in counter-clockwise order, when looking from the outside. This feature makes it possible to calculate the normal of the face using the formula, assuming  $A, B, C$  are face points



FIGURE 5.2. Subset of data used to test the mesh generator. Data comes from the micro-CT of a rat femur bone and was selected because of complexity of the resulting mesh.

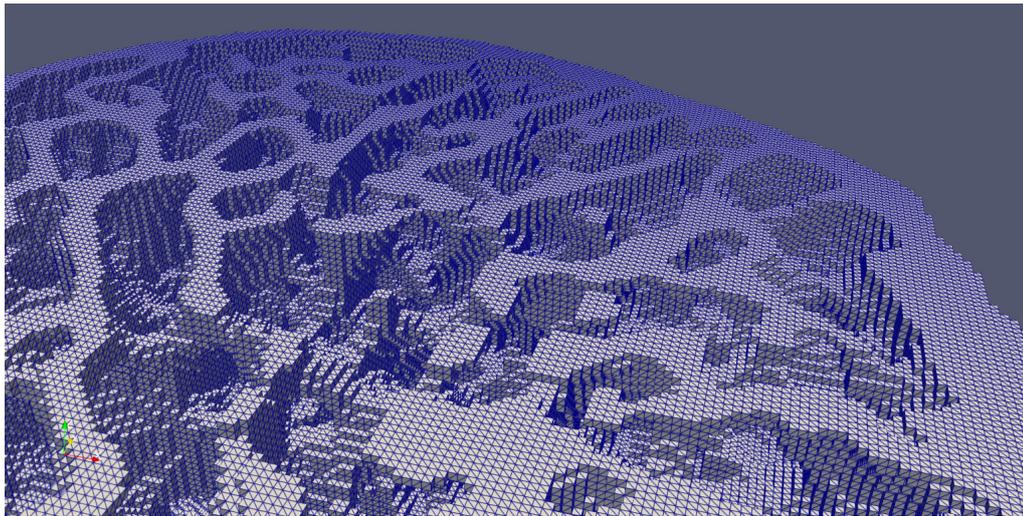


FIGURE 5.3. Mesh of the rat femur bone created from the micro-CT scan. Cubical structural elements, each containing 6 tetras are clearly visible.

in counter-clockwise order.

$$N(a, b, c) = \frac{\overrightarrow{(a-b)} \times \overrightarrow{(a-c)}}{|\overrightarrow{(a-b)} \times \overrightarrow{(a-c)}|} \quad (5.1)$$

The 'flipped normals' check finds faces that do not conform to this convention.

- The majority of surface mesh formats are unstructured. For example the most commonly used Standard Triangle Language (STL) format contains a list of vertices of triangular faces, Object File Format (OFF) contains separate lists of nodes and connectivity. This makes it possible for duplicate faces to exist in the mesh data. The 'duplicated faces' check searches for such issues.

- The 'duplicated points' check searches for duplicated data in formats when the points list is provided separately. It is important to note, that mesh points can be reported as duplicated if algorithm decides are close enough to cause numeric issues, even if displayed correctly.
- A proper mesh is a manifold – by definition each point on a surface locally resembles Euclidean space. In practice this check is implemented by finding edges shared by more than 2 faces. In the proper, watertight mesh each edge is shared by exactly two faces. Meshes with more than two faces sharing an edge are invalid and are reported by 'non-manifolds' check. There is also another kind of non-manifold mesh errors, so called 'bow-tie' errors when two parts of mesh are joined with just a single point. Detection of such points is time consuming and has to be enabled separately in settings.
- The 'degenerated faces' check finds faces with zero surface area.
- 'Face indices' check finds invalid indices in structured meshes.
- 'Self-intersections', as the name implies, finds pairs of faces intersecting each other.

Test mesh was analyzed with a mesh evaluator, the result is visible in the Figure 5.4. 856 non-manifold edges were found. One of such errors is in Figure 5.5. FreeCAD provides mesh repair capabilities, which were tested on the mesh. Results are in Figure 5.6. High mesh complexity prevents the automated holes filling algorithm from properly addressing mesh issues. It is possible to manually repair mesh by adding faces one by one, but doing this for all 856 non-manifold edges is not a viable task.

During tests viability of the mesh for manufacturing using classical approach with surface smoothing was also evaluated. As this is not the main objective of the research, visual assessment was performed. Laplacian Smoothing filter from the ParaView software was used. It was noticed that a high amount of smoothing is required to restore a smooth surface. Test results are visible in Figure 5.7. Mesh, produced by the current implementation of the mesh generator, cannot be easily smoothed by the Laplacian Smoothing algorithm. Number of iterations required to obtain a smooth surface distorts fine features of the mesh. Either better quality of the mesh or better smoothing algorithms are required.

## 5.2 Recommended changes to the mesh generator tool

After evaluation, the mesh generator was considered a viable solution for further research, but minor changes were recommended to improve overall mesh quality. To address identified issues changes in the implementation of the mesh generator were recommended. The first change was to extend generator capabilities to use more detailed structural elements than cubes. Use of partial elements, such as wedges or pyramids enables a smoother mesh surface. Additional shapes of structural elements do not interfere with structural analysis performed using the FEM system as new elements are properly connected with the rest of the mesh and have the same sizes as structural elements used to build full

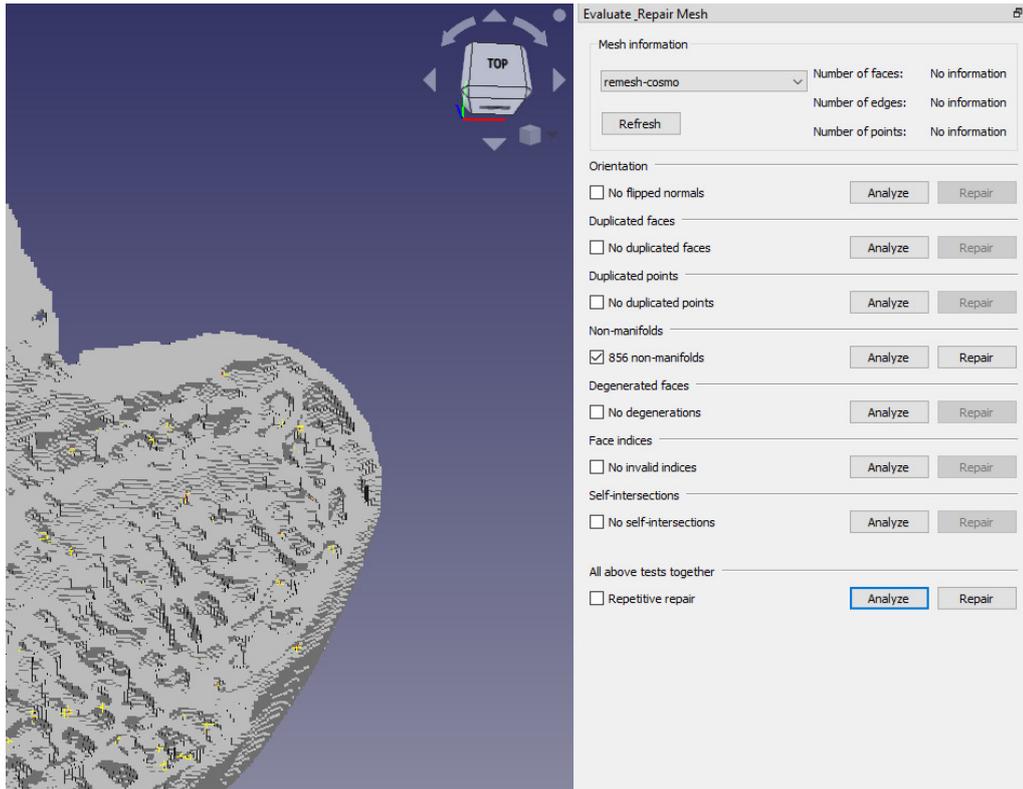


FIGURE 5.4. The result of the mesh analyzed with the FreeCAD system. Multiple non-manifolds were found.

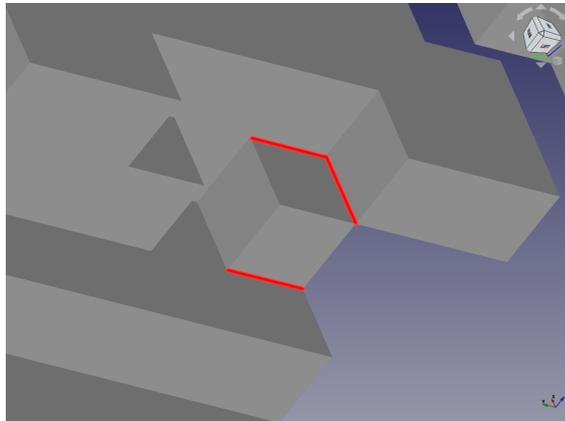


FIGURE 5.5. Mesh details with multiple non-manifolds highlighted in red.

cubes. As a side effect, a smoother surface should also increase the performance of the structural optimization process as a smooth surface enables better stress distribution on a surface without a stress concentration in concave areas of a rough mesh surface. Second recommendation is to include a non-manifold detection algorithm into the mesh generator. All available mesh repairing software removed non-manifold edges and filled created holes. This may lead to discontinuity in mesh and change the mesh topology as a result. To prevent this from happening, the recommendation is to fix non-manifold edges by adding, not removing mesh elements. It was observed that a biomimetic algorithm, once part is disconnected, is not able to connect it again but instead part is completely removed as it

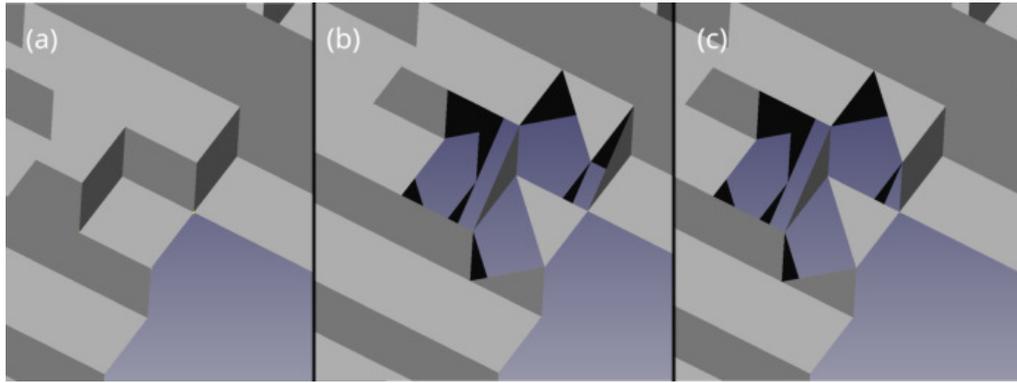


FIGURE 5.6. Showcase of the mesh repair feature in FreeCAD. (a) Mesh details with non-manifolds. (b) Mesh after using the repair option. Invalid edges were removed leaving open holes in the mesh. (c) Mesh after using the "fill holes" algorithm in FreeCAD, as recommended in the manual. Algorithm added just a single face, and wasn't able to correctly fix the mesh.

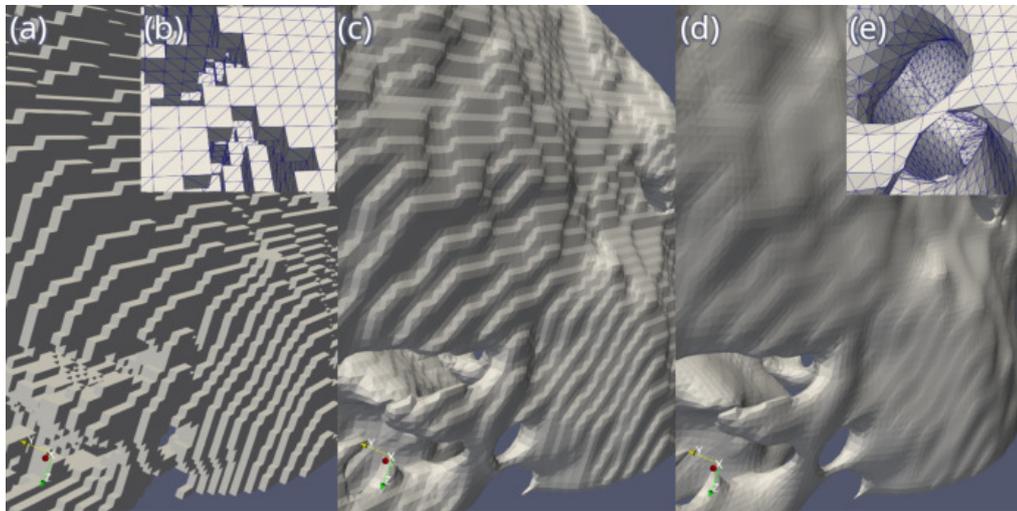


FIGURE 5.7. Smoothing of the mesh surface using the Laplacian Smoothing algorithm. (a) The input mesh. (b) Detail of the input mesh (c) After 200 iterations of Laplacian Smoothing. Rough surface is still noticeable. (d) After 500 iterations of Laplacian Smoothing. Rough surface is noticeable but smoothness can be considered sufficient. (e) Detail from (b) after 500 iterations of Laplacian Smoothing. Significant reduction of volume is visible.

no longer carries loads. Hence adding additional material is a safer approach to fix mesh issues and leads to higher quality optimization results. Detection of non-manifolds caused by elements connected by edges is trivial – requires counting the amount of faces at each shared edge and reporting number higher than 2. In practice if an edge with a single face is encountered the algorithm should stop immediately as such a case should never happen in properly generated mesh. Detection of single point non-manifolds required a dedicated algorithm. It can be noticed, that the properly constructed mesh can be represented as two dual, topologically equivalent graphs:

- Points connectivity graph – mesh points are graph nodes, mesh edges are graph edges,
- Faces connectivity graph – face central points are graph nodes, edges connect adjacent

faces.

When a mesh contains a single point non-manifold those graphs are no longer equivalent – point connectivity graph contains paths not reflected in faces connectivity graph. Moreover it can be observed that in the properly built mesh each faces connectivity subgraph created by selecting faces with a given shared edge contains exactly one cycle. Hence detection of non-manifold edges can be performed by creating a faces connectivity graph and for each point checking if there is exactly one cycle, as shown in Figure 5.8. In practice it is enough for a given subgraph to traverse it starting from an arbitrary face until returned to the initial point. If the amount of visited nodes is equal to the amount of nodes of subgraph the point is a valid mesh point. As non-manifold issues can appear in a mesh after addition of elements to fix detected errors the algorithm should repeat verification and fixing steps until all issues are corrected.

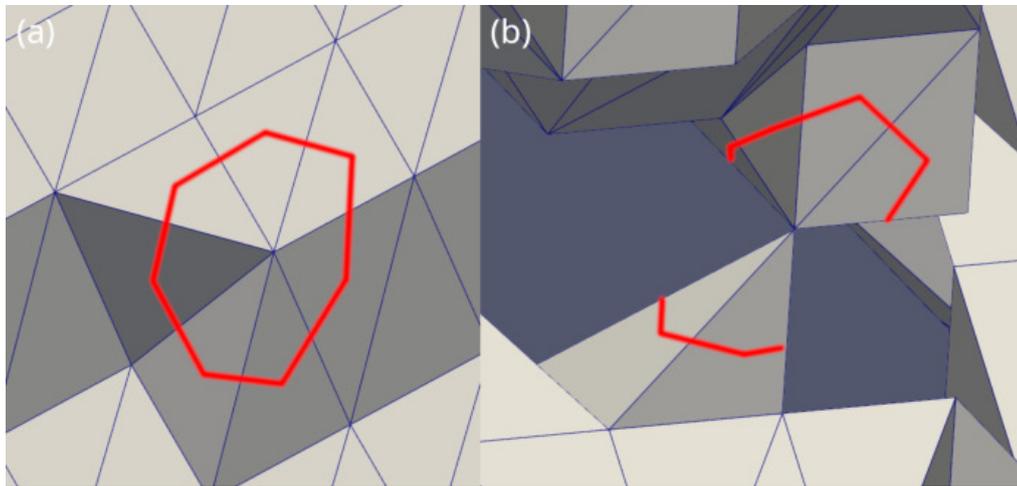


FIGURE 5.8. (a) In a correctly built mesh each faces connectivity subgraph created from faces sharing a given edge contains exactly one cycle. (b) Detail of a mesh with the single point non-manifold. When the faces connectivity subgraph is constructed for faces sharing given point, there exist two separate closed paths allowing detection.

## Chapter 6

# 3D Shapes Recognition using Hough Transform

Chapter describes feature recognition algorithm based on the Hough transform. It is the first, out of two, approach to the parametric model recognition researched and presented in this dissertation. Two-stage computer vision approach is applied to the finite element mesh to detect cylinders. Details of each algorithm step are provided and verification methodology for case studies is proposed.

Hough transform is a feature extraction technique commonly used in image processing [52, 53]. Initially designed for the line detection in 2D images, it can be adapted to work with other features and in 3D. It works by transforming data into parameter space, where each dimension represents one degree of freedom. Practical implementation uses accumulator space instead, which is a multidimensional array with each cell representing one potential set of result parameters. Line equation is  $y = ax + b$ , hence to detect lines two parameters must be estimated. For each point in the input data all lines containing that point are calculated and appropriate accumulator cells are incremented. Each accumulator cell contains votes for appropriate estimated parameters. Solution with the most votes is selected as a recognized feature. Example of an algorithm for recognition of lines in 2D is shown in a Figure 6.1.

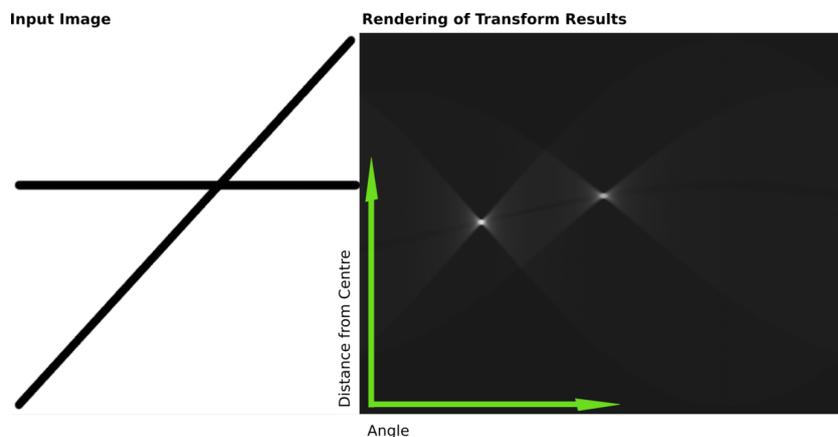


FIGURE 6.1. Example of the Hough transform for line detection. Left: Input data. Right: Hough space accumulator obtained for the data on the left. By Daf-de - Own work, CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=1121165>.

Hough transform is broadly used for industrial application. The main applications are computer vision and image recognition areas, but there are multiple uses in the area of virtual engineering and additive manufacturing. Hough transform can be used for CAD model detection and classification [54–56]. Depending on the task the algorithm is either used to detect features on pictures or scanned models. Fast detection of pre-defined features on pictures, makes it possible to use the Hough Transform algorithm for quality control using computer vision approaches [57–59]. Another application of Hough transform is detection of features in structural optimization results, either in 2D or 3D results [6, 60].

Structural Optimization results visually resemble trusses containing multiple cylindrical objects. This suggests that Hough transform can be used to detect those cylindrical features. As Hough transform requires accumulator space for each degree of freedom of recognized feature it is impractical to create such data structure for single-pass recognition. Instead the approach described in [61–63] was used. Following implementation is based on the algorithm described in [62].

## 6.1 Orientation Estimation

The first step of the algorithm is finding the direction of the cylinder in input data. Implementation used in this step is based on an observation that for a cylinder all normals of the side surface form a great circle on the Gaussian sphere. This great circle results from the intersection of the unit sphere with the plane passing through the origin. The normal vector of this plane is the same as the cylinder axis. Accumulator stores votes for direction of this normal. For each mesh face in the input data set the algorithm adds to the accumulator votes for all possible normals of cylinders that can contain a given face (Figure 6.2). For each normal vector  $N$ , votes are calculated by finding all points of parametric 2D circle given by the Formula 6.1.

$$x = \cos(t), y = \sin(t) \quad (6.1)$$

Equation is solved for  $t$  from the  $[0, 2\pi)$  range. Each value obtained is treated as an unit vector  $V = (x, y, 0)$  and transformed into normal plane to the vector  $N$  using rotation matrix. To create rotation matrix first angle  $\alpha$  and direction  $u$  of required rotation is calculated according to the Formula 6.2.

$$Z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \alpha = \arccos((Z \times N) \cdot Z), u = (Z \times N) \quad (6.2)$$

Calculated angle and direction provide rotation of unit vector  $Z$  along the  $Z$  axis to the given normal vector  $N$ . Rotation matrix is calculated based on two values using Formula 6.3.

$$R = uu^T + \cos(\alpha)(I - uu^T) + \sin(\alpha) \begin{pmatrix} \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \end{pmatrix} \quad (6.3)$$

The rotated unit vector  $V'$  is created by applying rotation matrix to the vector  $V$  converted into spherical coordinates  $\theta, \phi$  which are used directly as position in a Hough SpaceAccumulator according to Formulas 6.4 and 6.5.

$$\theta = \text{acos}(V'_z) \quad (6.4)$$

$$\phi = \frac{V'_y}{|V'_y|} \text{acos} \left( \frac{V'_x}{\sqrt{V_x'^2 + V_y'^2}} \right) \quad (6.5)$$

Unlike the cited paper, where recognition was based on a points cloud this algorithm works on triangulated surfaces directly so there is no need of estimating normals. Normal for a triangular face with points  $a, b, c$  can be calculated using the Formula 6.6.

$$n(a, b, c) = \frac{\overrightarrow{(a-b)} \times \overrightarrow{(a-c)}}{|\overrightarrow{(a-b)} \times \overrightarrow{(a-c)}|} \quad (6.6)$$

After adding all votes to the Hough accumulator maximal value, corresponding to the axis of a cylinder is located. As shown in Figure 6.3 due to both numerical error and mesh imperfections, the area corresponding to the axis is not a single pixel in the accumulator, but instead is a blurred area. To compensate for this phenomenon, instead of scanning for simple maximal values, the average value in a wider area is calculated using a morphology kernel. Tested implementation uses a 21x21 pixels square kernel with equal weights.

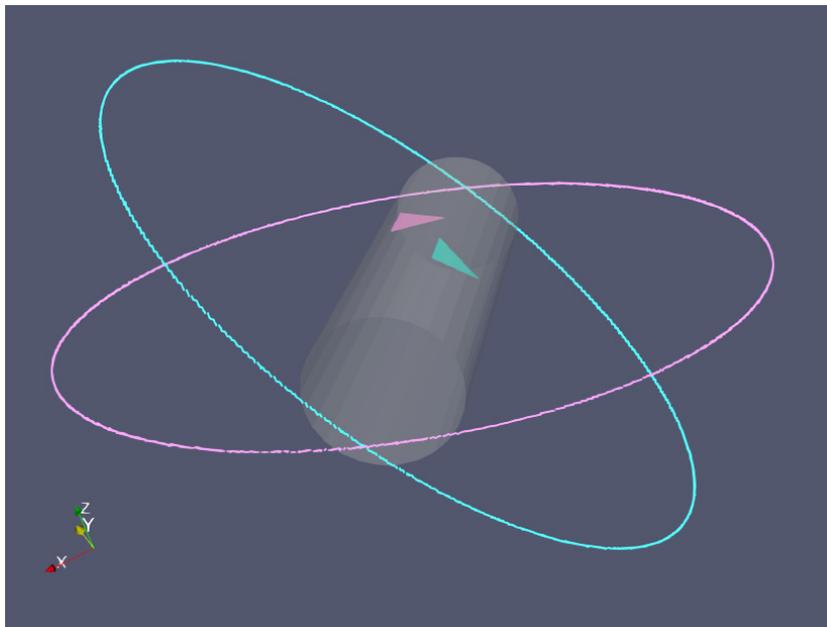


FIGURE 6.2. Two faces of a cylinder surface with corresponding circles in Hough space. Each circle corresponds to all possible axes of cylinders that include a given face. The color of the face corresponds to the color of the circle generated of its normal. Intersection of two circles corresponds to the actual axis of the input cylinder. Data comes from the actual algorithm output. Faces were obtained from triangulation using gmsh software with the ‘Frontal Delaunay’ generator [64, 65].

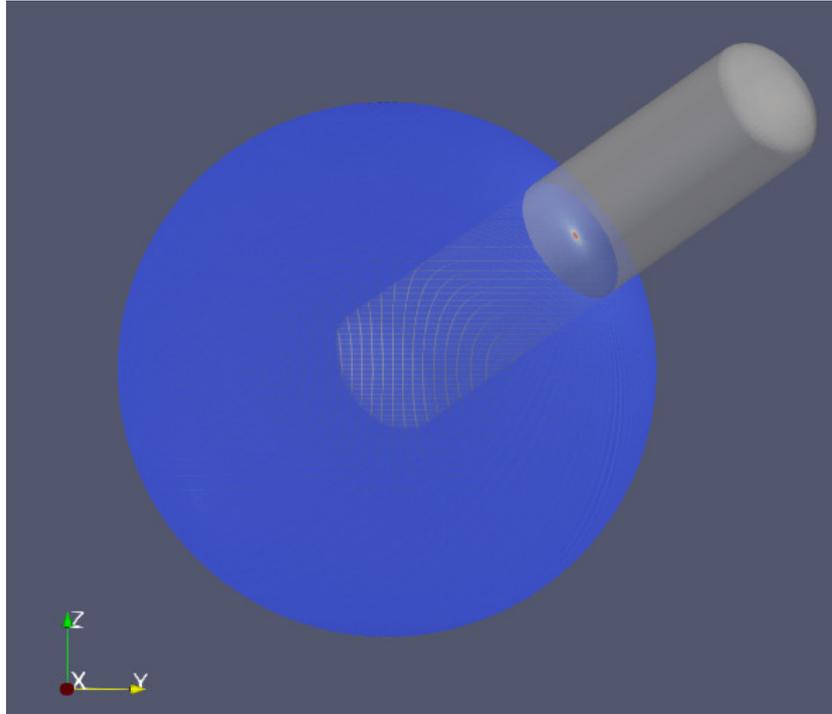


FIGURE 6.3. Hough space accumulator for the mesh generated with the MeshAdapt algorithm [66] with an overlaid input cylinder, shrunk down to match the size of the Gaussian sphere. Point with maximal accumulator value corresponds to the cylinder axis.

## 6.2 Position and Radius Estimation

Second step of the algorithm, after the cylinder axis is estimated, is cylinder position and radius estimation. First faces of the input mesh are filtered to obtain the cylinder surface set. For each face the angle between a face normal and the cylinder axis is calculated. Face, to potentially be part of the estimated cylinder, must have the normal vector perpendicular to the cylinder axis. To account for errors a small margin is used so faces with this angle between  $(0.95\pi, 1.05\pi)$  are selected. As this step of the algorithm operates on points, all faces are converted into points. Those points are projected onto the plane normal to the cylinder axis  $V$  using planar projection matrix used in a software rendering shadow mapping [67] shown in Formula 6.7. Points projected this way are shown in Figure 6.4.

$$M = \begin{bmatrix} -V_x^2 + V \cdot V & -V_y V_x & -V_x V_y & 0 \\ -V_x V_y & -V_y^2 + V \cdot V & -V_y V_z & 0 \\ -V_x V_z & -V_y V_z & -V_z^2 + V \cdot V & 0 \\ 0 & 0 & 0 & V \cdot V \end{bmatrix} \quad (6.7)$$

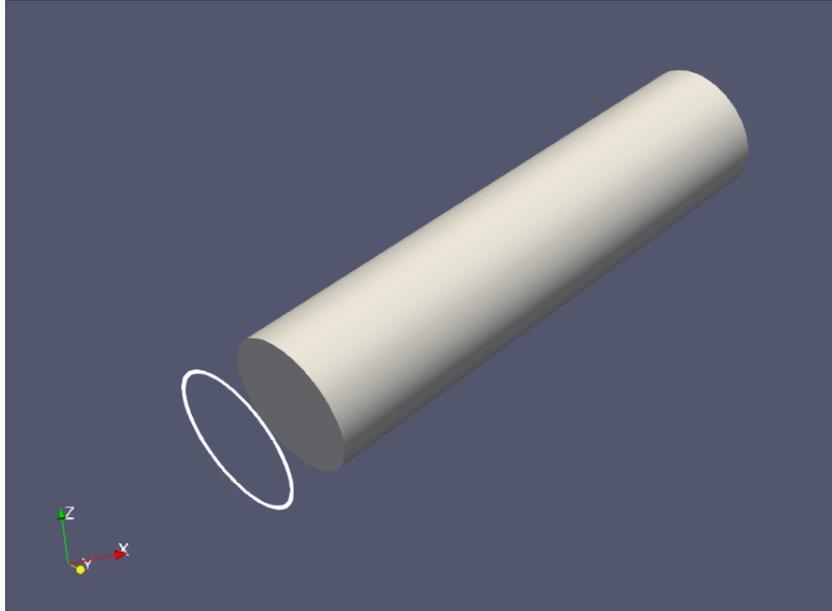


FIGURE 6.4. Cylinder surface points projected to the plane normal to the cylinder axis.

To simplify further calculation, data is rotated so that the projection plane normal is aligned to the Z axis of the coordinate system, reducing the problem space to two dimensions rotation matrix provided by Formula 6.2. Now the problem of estimating position and radius of the cylinder is classical circle finding problem [68, 69]. Standard Hough Transform approach was selected for the initial implementation despite more efficient algorithms being available. Implementation was created to evaluate the feasibility of using the Hough transform for the recognition of features of the structural optimization output and the simplest approach was considered sufficient. The approach uses a three dimensional Hough space accumulator with dimensions representing radius, x coordinate and y coordinate of estimated circle. The accumulator is filled for each point with votes for centers of all possible circles that contain a given point. In practical implementation accumulator space is a stack of 2-dimensional accumulators with a separate accumulator for each circle radius. For each point in the input data, in each 2-dimensional accumulator, a circle of appropriate radius is drawn using a standard approach from computer graphics - the Bresenham's Circle Drawing Algorithm [70]. In the accumulator corresponding to the right radius all drawn circles will converge giving maximum accumulator value which is located and the position and radius can be directly obtained out of it (Figure 6.5).

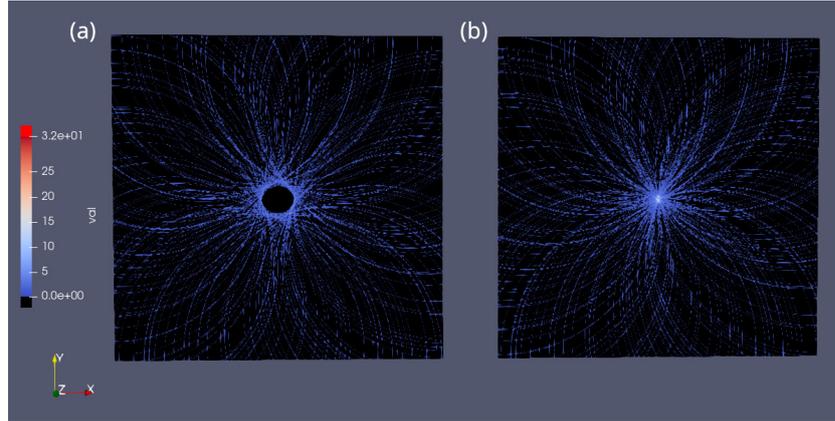


FIGURE 6.5. Two slices of Hough space accumulator used for the position and radius estimation. Accumulator is filled using a random subsample of 5% of the input data. (a) The slice corresponding to the radius of 4.5mm. (b) The slice corresponding to the radius of 5.0mm. When radius is correctly estimated, all circles in the accumulator converge in a single point, which corresponds to the right radius and position.

### 6.3 Shape Ends Positions Estimation

Third step, to fully capture all parameters of the cylinder, is the estimation of positions of two end points of the cylinder. As the axis position and orientation is known, this task is reduced to one dimensional local search to find both points. Points that took part in voting in the previous step are used and points that are not inside the identified cylinder are removed. Coordinates are transformed using a rotation matrix to align the cylinder axis along the Z axis of the new coordinate system. The X and the Y coordinate both ends are now the same as the axis. The minimal and the maximal value of Z coordinate across all cylinder's points are selected as positions of cylinder ends. Final coordinates are transformed back into original coordinate space and returned.

### 6.4 Post-recognition step

The algorithm presented in [62] runs multiple Position And Radius Estimation steps for each local maximum identified during the first step. Proposed implementation works in an iterative manner instead. After each step all faces inside the identified cylinder are removed and the algorithm runs again using the obtained subset as an input. Removal of processed data increases accuracy of the algorithm when input data contains cylinders with closely aligned axes. Hough transform used for line detection described in [6] exhibited such behavior and accumulator size was increased to increase detection resolution. Algorithm there was tested on a synthetic data set. For real data, coming from the structural optimization software, noise is expected to degrade detection quality. Accumulator resolution increase doesn't not help when data overlaps due to noise.

## 6.5 Implementation notes

It is important to note, that the implementation used for following tests contains issues that prevent it from being used outside scope of tests. The goal of the research was to evaluate the feasibility of the presented approach for the feature recognition of structural analysis results. Identified and ignored problems are:

- Transformation of spherical coordinates into a two-dimensional array using naive  $(x, y) = (\phi, \theta)$  formula results in two singularity points at  $\phi = \pi$  and  $\phi = -\pi$ . This makes it impossible to correctly detect cylinders with an axis close to  $\phi = \pi$ . The original paper doesn't mention this issue.
- Second problem with the used mapping is non-uniform accumulator density - polar regions have denser coverage than equatorial areas. This issue doesn't affect the voting process. The algorithm, during the accumulator filling step, defects all candidate points and increases appropriate accumulator values. On the other hand the kernel used for votes averaging averages data from a wider area on the equator than near poles. The original paper finds max value in accumulator using simple max operator so this issue doesn't exist there.
- Full unit sphere is used as an accumulator for the cylinder axis orientation finding step. Half of the accumulator size would be sufficient - the sphere is symmetrical as cylinders axes intersect it in two points. Future implementations should have memory by using the right size of accumulator.
- The original paper mentioned data was filtered out to exclude planes because 'big planar areas interfered with cylinder axis estimation'. In researched implementation no effort to reproduce this was made. Instead test data without such features was used. This is the reason why disks at the end of cylinders in test data were replaced with domes.
- Circle detection was implemented using inefficient Standard Hough Transform. More robust approaches, such as [71], are available and should be used in future research.

## 6.6 Algorithm Validation Methodology

The output of the feature recognition algorithm is a set of cylinders. Each cylinder is described by 3 parameters: two points, A and B and radius R. To measure quality of the solution measures are calculated, given the source cylinder is described by two points  $A_s$ ,  $B_s$  and radius  $R_s$ . The first set of measures are simple differences – Formulas 6.8, 6.9, 6.10. A proper estimation of cylinder axis followed by improper estimation of cylinder ends location may lead to similar delta values as wrong axis angle estimation, despite the second case can be considered as worse than the first one. Correction of cylinder length when proper axis is known requires local search in single dimension performed for both ends, correction of the axis requires the local search of both points position, hence scanning all 6 degrees of

freedom. To estimate quality of axis estimation three additional measures are used. *DistErr* is a distance between cylinder end points and axis of source cylinder – Formulas 6.13, 6.14. *AxisErr* is an angle between a source cylinder direction vector and a reconstructed cylinder direction vector – Formula 6.15.

$$\Delta A = |A - A_s| \quad (6.8)$$

$$\Delta B = |B - B_s| \quad (6.9)$$

$$\Delta R = |R - R_s| \quad (6.10)$$

$$Dir_s = \frac{B_s - A_s}{|B_s - A_s|} \quad (6.11)$$

$$Dir = \frac{B - A}{|B - A|} \quad (6.12)$$

$$DistErr(A) = |(A - A_s) - ((A - A_s) \cdot Dir_s) Dir_s| \quad (6.13)$$

$$DistErr(B) = |(B - A_s) - ((B - A_s) \cdot Dir_s) Dir_s| \quad (6.14)$$

$$AxisErr = \frac{180}{\pi} \text{acos}(Dir \cdot Dir_s) \quad (6.15)$$

## Chapter 7

# Automated, Skeletonization-Based Truss Recognition

In this chapter recognition algorithm based on the mesh skeletonization algorithm is described. Approach uses skeletonization to determine the initial geometry, which is refined using an evolutionary algorithm. Details of each algorithm step are provided and verification methodology for case studies is proposed. Presented research was published [72].

The recognition algorithm uses few distinctive steps, as seen in Figure 7.1. The input of the algorithm is both the input data and the resulting mesh of the structural optimization system. The sample input for the algorithm, created using FreeCAD software, is provided in Figure 7.2. The first step is the extraction of the area of interest. The result of the structural optimization contains the whole optimization domain, which may include areas excluded from optimization. Those areas are both positive and negative exclusions, that is, areas which contain material that must not be removed, such as attachment points and areas which must not contain material, such as holes for screws or those occupied by elements that are not a subject of the optimization. Areas which contain material and are excluded from optimization are removed from the input mesh—a parametric model for those already exists as an input for the algorithm so there is no need to re-import it to the CAD system. The definition of areas excluded from optimization is taken from the input of the structural optimization system. The second step is skeletonization. The mesh skeleton is obtained, processed and used in the next step to generate the initial solution. The solution is then iteratively refined using a heuristic optimization approach. The refined solution is converted to a format that can be imported to the CAD system—FreeCAD.

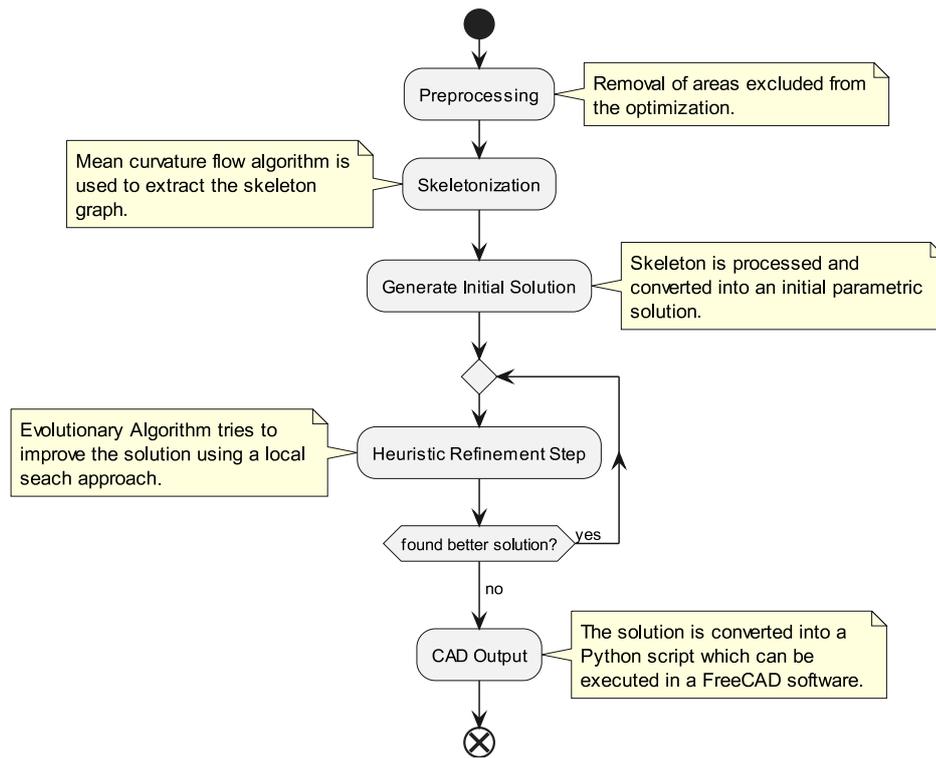


FIGURE 7.1. Activity diagram of the feature recognition algorithm. Preprocessing and skeletonization steps lead to an initial solution, which is refined using a heuristic approach.

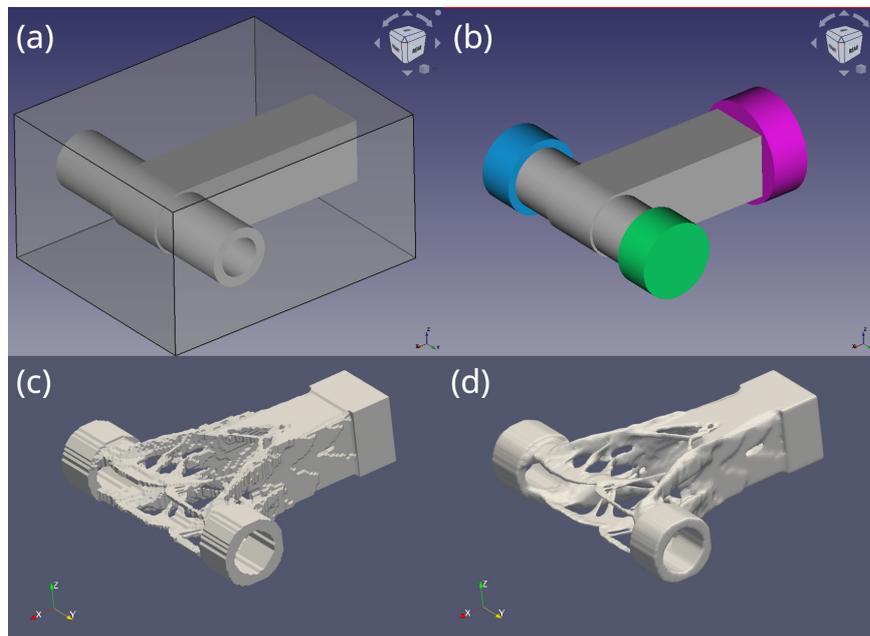


FIGURE 7.2. Example of feature recognition algorithm input. (a,b) Input of the structural optimization algorithm, as seen in the CAD system. (a) Optimized object and optimization domain boundary. (b) Optimized object and three boundary condition areas, which in this particular case are also areas excluded from optimization. Bottom: structural optimization output. (c) Raw surface mesh. (d) Surface mesh smoothed, only for visualization purposes.

## 7.1 Preprocessing

The output of the structural optimization is a triangulated mesh, ready to use by the truss recognition algorithm. The first step is the removal of areas defined in the optimization task as being excluded from the optimization. Those areas, in practice, contain attachment points that must be preserved throughout the whole process and are not subject to either structural optimization or feature recognition.

The removal of the excluded areas is performed by the structural optimization software by generating a mesh consisting of all the points that do not belong to the excluded areas. Using the same generator as used for the structural optimization results is important to preserve the mesh points density, which is also important in the next steps of the algorithm. A constant mesh vertex density improves the reliability of the mesh skeletonization and is required later by the refinement step to calculate the score of each candidate. Initially, an algorithm provided by CGAL library was used, but re-meshing applied by it altered the mesh density which, in turn, decreased the quality of the next steps of the feature recognition process. A comparison of both approaches is illustrated in Figure 7.3.

## 7.2 Skeletonization

The mesh skeletonization is performed using the mean curvature skeleton algorithm [73, 74]. Unlike other approaches, such as the algorithm used in [23], the mean curvature skeleton output is a graph representing the mesh skeleton and there is no need for additional processing. The skeletonization algorithm works really well with a biomimetic optimization output despite the jagged surface of the resulting mesh. There are no disconnected areas or undesirable skeleton curves. Such artifacts are often presented in skeletons obtained by techniques based on morphological filtering, as seen in [9, 75]. As shown during preprocessing, the skeletonization algorithm is sensitive to mesh density. The resolution of the skeletonization can be increased by increasing the mesh density using a loop subdivision algorithm [76]. The performed tests showed no improvement of the initial solution's quality when using higher resolution meshes, but the skeletonization time increased greatly. Hence, the mesh returned by a mesh generator from a structural optimization system is used as-is for the skeletonization algorithm input. Sample meshes with matching skeletons are shown in Figure 7.4.

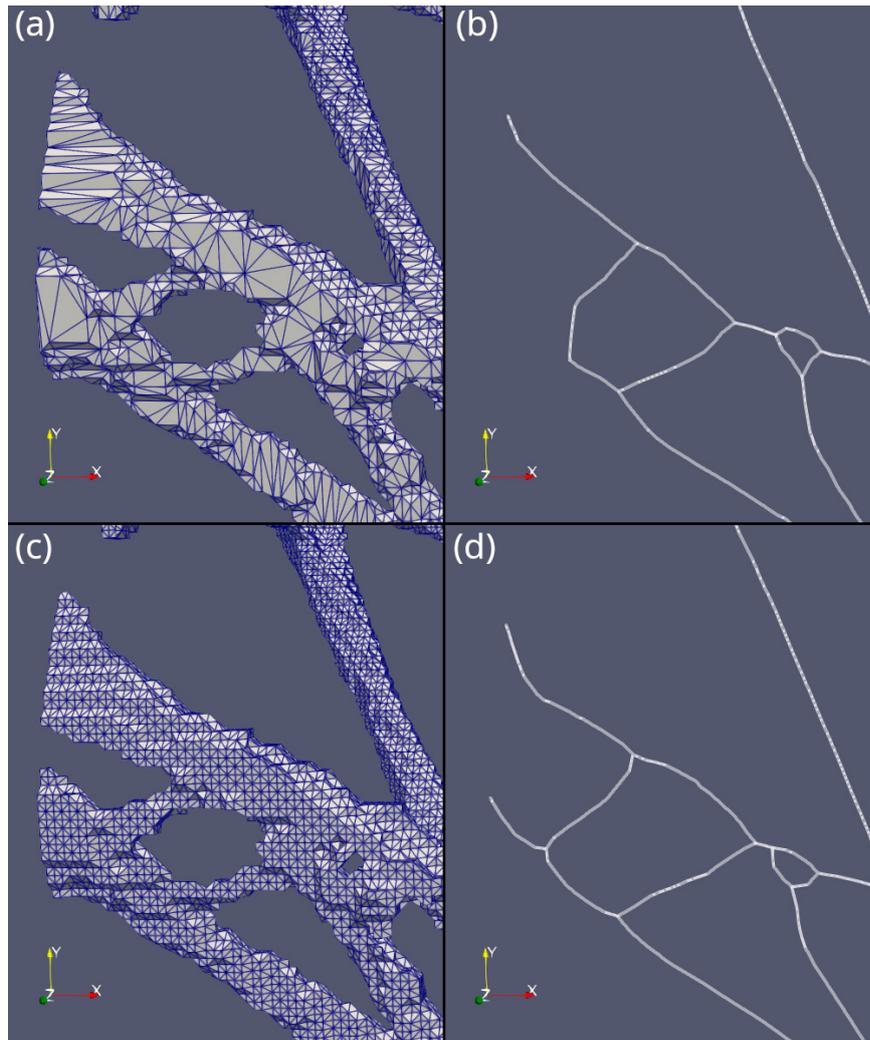


FIGURE 7.3. Impact of mesh density on skeletonization algorithm. (a) Mesh preprocessed with CGAL algorithms to perform removal of areas excluded from optimization. (b) Skeleton generated for (a) mesh. One of the structural elements completely collapsed. (c) Mesh with removed areas excluded from optimization using mesh generator from structural optimization system. Mesh retained vertex density. (d) Skeleton generated for (c) mesh.

There was no attempt to improve the skeleton alignment inside the mesh. Although this would improve the initial solution quality, for proof of concept work, this part is considered redundant. The refinement step, performed next, should yield similar results.

### 7.3 Skeleton Processing

The graph representing the skeletonized mesh is processed to obtain the initial solution. This process has a single configuration variable: the minimum path length. In the first step of the process, all paths between the graph's loose nodes (with a single edge) or intersection nodes (with more than two edges) are calculated. Those paths are either replaced with single edges, or, if the path curvature is high enough, with a series of edges approximating the path. A biomimetic approach used as an input of the recognition algorithm yields a porous and unstable structure in areas where the mesh resolution is insufficient to represent the fine microstructure present in the analytical solution, as shown in Figure 7.5. This

phenomenon justifies filtering out such porous structures in the initial solution. Hence, all edges shorter than the given minimum path length are collapsed. This also enables some degree of solution resolution control—a high value of minimum path length simplifies topology by collapsing the fine details. Sample data illustrating the whole process are illustrated in Figure 7.6. Finally, the graph obtained this way is converted into the initial solution by assigning the default radius to each graph edge.

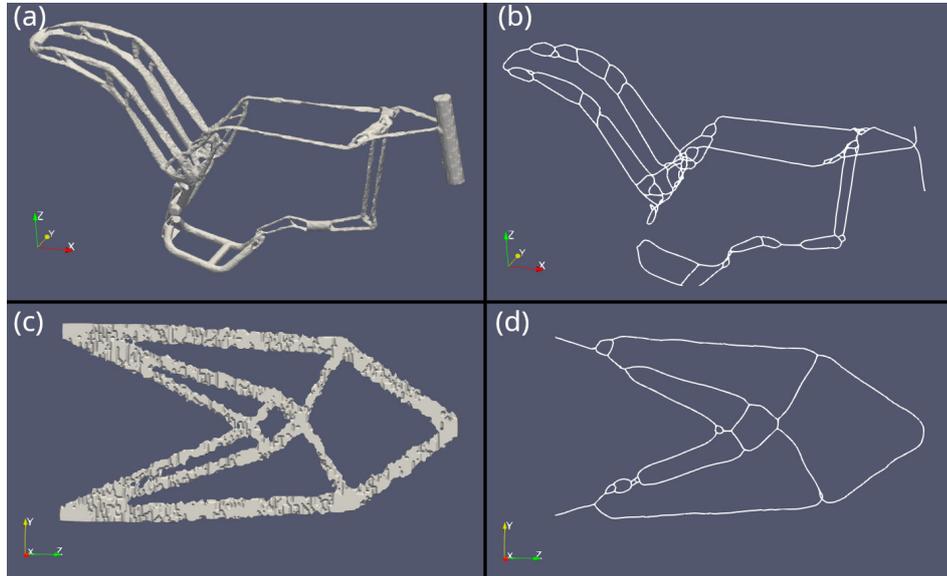


FIGURE 7.4. (a,c) Example output meshes from topology optimization system. (b,d) Skeletons generated for corresponding meshes.

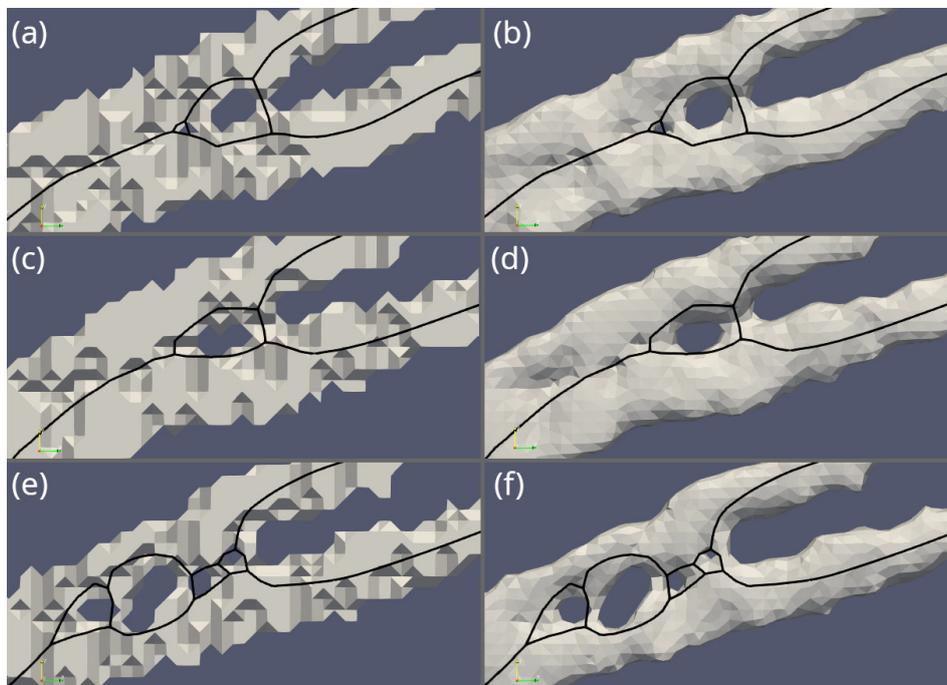


FIGURE 7.5. Justification for skeleton filtering: single detail of the structural optimization output and corresponding skeleton for 3 consecutive optimization steps. (a,c,e) Details of structural optimization output with skeletons drawn over. (b,d,f) Details on the left with smoothing applied, for visualization purposes only, with skeletons drawn over. Detail represents an unstable region of the optimization domain—the porous structure appears and disappears because a mesh density is too low to converge to the stable state.

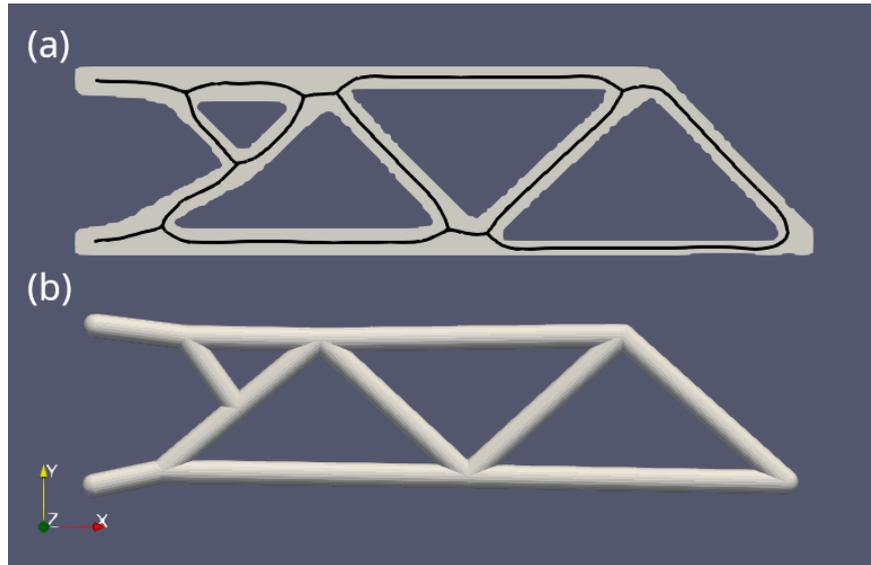


FIGURE 7.6. (a) Optimized cantilever beam. Mesh was created out of a 2D image by extruding it. Skeleton generated for the shown mesh is drawn over. (b) Initial solution generated from processed skeleton. Two skeleton edges were below the minimum length threshold and collapsed. Bottom-right node of the solution graph was generated by splitting an edge of the skeleton graph.

## 7.4 Heuristic Refinement

The initial solution obtained from the skeletonized and processed output needs further refinement. This refinement is performed using an evolutionary algorithm that iteratively improves the solution. Unlike the biomimetic structural optimization, the iterative algorithm is based on a theoretical mathematical model proved to be valid, and the proposed solution is heuristic and created experimentally. The algorithm is designed to be fast, and the trade-off is reduced accuracy.

The evolutionary algorithm requires the definition of the Fitness Function to calculate a score for each candidate and select the best one. The Fitness Function must be quick to compute and at the same time accurate to enable visiting many candidates in a short time. The optimal solution of a feature recognition algorithm is a set of features that completely fill the volume of input data and do not exceed it. Scoring based on volume requires the time-consuming operation of intersecting the input mesh with the scored solution to find differences. The optimization of this approach uses a mesh discretized using a slice-based approach and comparing discrete sets of points. This approach has  $O(n^3)$  complexity, so a more efficient but less precise approach was used. As the algorithm operates on the surface, the not-volumetric, mesh, surface-based approach was also used for the Fitness Function. Two sets of points are calculated: control points and guide points. Control points are points on a mesh surface that are used to determine how close a solution's surface is to the input surface. A set of guide points is obtained by the morphology dilation of the input mesh. The size of the dilation is an input parameter of the algorithm. Details of the mesh surrounded by the guide points are shown in Figure 7.7.

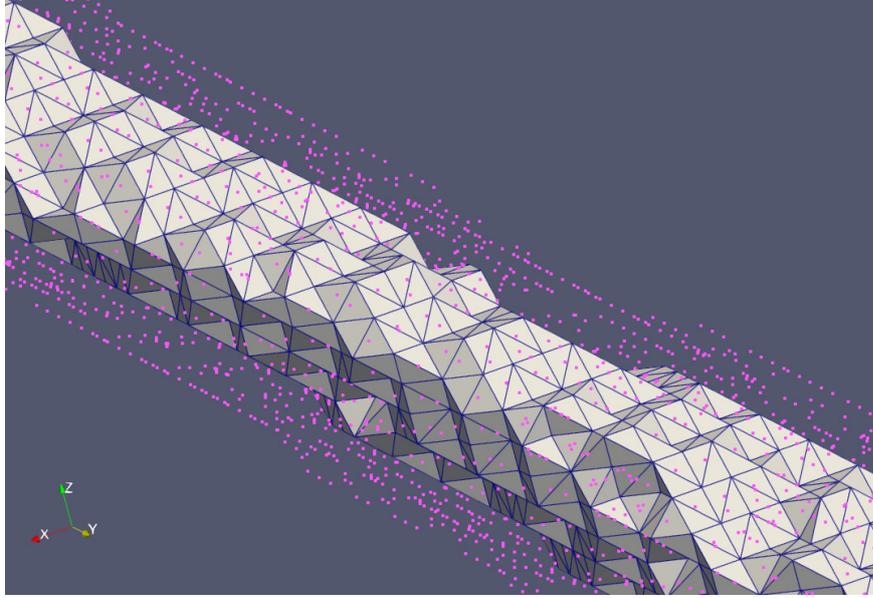


FIGURE 7.7. Detail of the input mesh surrounded by guide points (pink). Fitness Function yields positive score for aligning solution to the mesh surface and gives penalty for including guide points.

The Fitness Function is calculated for a solution consisting of the set of elements,  $E$ , the given set of control mesh points,  $P$ , and guide points,  $G$ .  $Surface\_distance(p, e)$  is a function that returns the Euclidean distance between element  $e$ 's surface and point  $p$ .  $is\_inside(p, e)$  is 1 if the point  $p$  is inside geometry  $g$ , and is otherwise 0.

$$S_{positive} = \sum_{p \in P} \min_{e \in E} surface\_distance(p, e)^2 \quad (7.1)$$

$$S_{negative} = \sum_{g \in G} \min_{e \in E} -(surface\_distance(g, e) * is\_inside(g, e))^2 \quad (7.2)$$

$$Fitness = \alpha S_{positive} + \beta S_{negative} \quad (7.3)$$

The solution is scored positively for the matching input surface and is penalized for enveloping guide points and taking more volume than required. Both positive and negative scores are aggregated using weights,  $\alpha$  and  $\beta$ , used to equalize imbalances in the size of two sets. For tests,  $\alpha = 1$  and  $\beta = 50$  were used. To increase performance, for each evolutionary algorithm iteration, a subset of both sets was generated. A total of 20% of mesh points and 4% of guide points were randomly selected. The size of the guide set is five times smaller than the size of the control points so, effectively, the weight of the negative score was ten times higher than the positive. It is important to note that this approach works well because the input mesh, obtained from the mesh generator, which is part of the structural optimization algorithm, is built from regularly spaced elements. Without this feature, an extra algorithm step would be required to ensure that the mesh vertices are equally spaced and uniformly cover the mesh surface.

The evolutionary algorithm works iteratively, creating a set of candidates using random sampling of a solution neighborhood space using two available operations. In the descriptions,  $\mathcal{N}(\mu, \sigma^2)$  is the function returning the normally distributed random value. The first

operation is moving one randomly taken solution graph node by random vector

$$v = \begin{bmatrix} \mathcal{N}(0, \sigma_{dir}^2) \\ \mathcal{N}(0, \sigma_{dir}^2) \\ \mathcal{N}(0, \sigma_{dir}^2) \end{bmatrix} \quad (7.4)$$

$\sigma_{dir}$  is a parameter of an algorithm. Second is the changing radius of the random cylinder by  $\mathcal{N}(0, \sigma_r^2)$ .  $\sigma_r$  is a parameter of an algorithm. Each candidate solution is created by applying random operations 1 to 5 times. A moving graph node has twice the probability of being applied than a changing radius. All candidate solutions are scored using a Fitness Function and the best one is returned. A single iteration visits 5000 solutions. The stop criterion is exceeding the pre-set number of iterations or there being no improvement after 20 iterations.

Although the current implementation reconstructs the optimized object using cylinders, the refinement step was designed to work with any part. A framework built in this way allows an easy change of structural element to, i.e. square beam, or the use of curved elements. The cylinder was chosen for proof of concept work because it has fewer degrees of freedom than other structural elements.

## 7.5 Postprocessing and CAD Integration

The refined mesh is converted into a format that can be imported into CAD software. FreeCAD was selected because of a non-proprietary license and extensive documentation. The results are converted into a Python script that uses FreeCAD's API [77] to create parts consisting of cylinders.

Additionally, the export script creates spheres at each node to ensure a proper connection between cylinders, as shown in Figure 7.8. The radius of each sphere is 110% of the biggest radius of the element connected to the node. The extra margin was added to avoid numeric errors reported by CAD software when fusing elements with a similar radius to produce the final mesh. Note that those spheres are only added in the final step and are not part of the 'main' geometry. Hence, to simplify the calculation of the Fitness Function, it was decided to ignore those extra parts during the refinement step, despite the fact that they contribute to the overall volume and mass of the model.

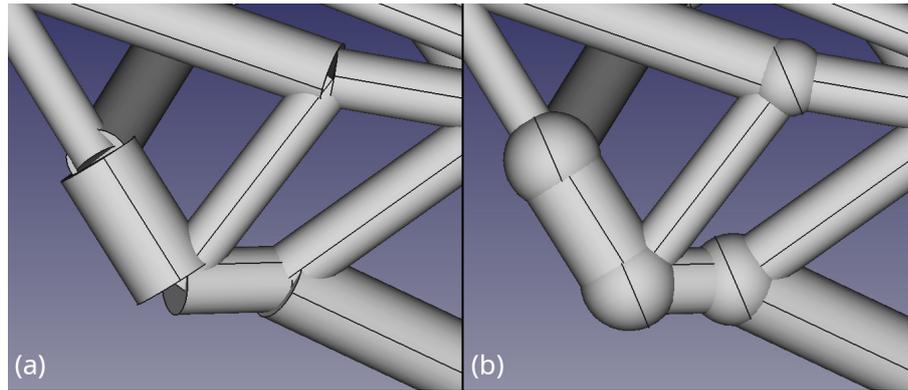


FIGURE 7.8. (a) Extracted features exported to CAD as cylinders. Partial connections between elements are visible. (b) The same features exported as cylinders with additional spheres at the ends. All elements are fully connected.

An alternative output is the implementation of producing a set of IGES (Initial Graphics Exchange Specification) format files, which provides a vendor-neutral way of importing object information into CAD systems.

## 7.6 Solution Verification—Methodology

The Fitness Function defined for the evolutionary algorithm used during refinement step cannot be used to measure the quality of the solution. The Fitness Function is used to heuristically compare solution candidates but does not have the information if the mechanical properties of the solution match ones of the input mesh. Solution verification is performed by computing the same measures for two objects—the structural optimization result, which is also an input for the feature recognition system, and the reconstructed parametric solution, exported from the CAD system as an STL mesh.

The first compared measure is the total mass of an object. Objects reconstructed in CAD should not have a bigger mass than the source objects, so the mass of the object after the structural optimization is compared with the mass of the reconstructed object exported from the CAD system. The mass is calculated using the density of the material and the volume of the mesh. The density is an input parameter of the optimization process and the volume is calculated using a volumetric mesh generator, which is part of the structural optimization system. The volume of all constituent mesh cells is summed to get the total mesh volume.

The surface energy value, which is equalized during structural optimization, is selected in such a way that the von Mises stress does not exceed the critical value. Fulfilling the von Mises yield criterion guarantees structural integrity under the load of the optimization results. A second test is performed to prove that the von Mises stress in a reconstructed object does not exceed the maximum von Mises stress present in the structural optimization results. To obtain a stress distribution, the object is exported from CAD and loaded back into the structural optimization system. The exported surface mesh is first converted into a volumetric mesh and then into Finite Elements Method input data. A static elasticity calculation is performed using FrontISTR software [48, 78]. The maximal value of the von

Mises stress is obtained and the stress distribution is analyzed. Together with the values of the von Mises stress, the displacement data were obtained. Using mesh displacements, the maximum deflection under load is calculated. Despite it providing no extra information compared to von Mises stress, the maximum deflection is a value often used in engineering analysis, so it is also presented. In case of multiple load scenarios, tests must be performed for each scenario independently.

The presented solution verification has been performed for both cases described below.

## Chapter 8

# Application of the parametric model in the hybrid approach

The research described so far was mainly focused on the algorithmic aspects of parametric model building. In this chapter research takes a different direction and research is focused on the industrialization potential of the researched parametric model. Practical knowledge obtained during research projects is used to analyze demands. Based on that an innovative workflow that uses a hybrid approach is proposed. In the holistic view of the mechanical design process, structural optimization breaks the flow centered around the CAD software. Finite elements mesh from the structural optimization system cannot be easily altered. Mesh morphing operations are possible and there are dedicated tools allowing it, such as Autodesk Meshmixer [79]. With a rapid spreading and an advancement of additive manufacturing technologies one may consider a CAD processing, required for the manufacturing preparation, redundant or even obsolete. Virtually every additive manufacturing machine requires a dedicated software, which translates job into a set of machine codes for a device. Majority of software is able to accept the input data in a STL format, which became *de facto* standard interchange format in an additive manufacturing area. This means a mesh obtained from the structural optimization system no longer needs the parametric model, as a surface smoothing is the only processing required to prepare the model. In practice, however, limitations of the manufacturing techniques require the proper model processing step. Moreover, the proper mechanical design requires multiple prototype stages, where different manufacturing techniques are used, with different limitations. Each requires dedicated model preparation. The inability to adjust parametric models makes work to prepare the mesh for manufacturing laborious. Manual mesh sculpting operations require experience and repeatability of the work is low – every software operator will do this in a slightly different way. The parametric CAD model, on the other hand, allows altering the model in a standard way, leading to consistent results. Changes on a base parametric model can be applied independently to accommodate multiple manufacturing techniques with different limitations. The more complex the parametric model is, the better it reflects intrinsic complexity of structural optimization results. The trade-off is an ability to introduce changes. On the one end of the spectrum are parametric models created with techniques using cubic B-spline surfaces. With such a mathematical model it

is possible to perfectly reflect the results of the structural optimization. On the other hand introducing changes in a model is as difficult as with finite elements mesh and requires considerable experience. The opposite philosophy is embraced by approaches reconstructing source models from a set of primitive geometries. Beam model being extreme. Loss of details is offset by an ability to easily change the parametric model. The approach presented in previous chapters belongs to this family of approaches. Reconstructing the model as a truss leads to an inevitable loss of details, but the parametric model, after import to the CAD system, is convenient to change. In the BioniAMoto project a biomimetic structural optimization system is used to prepare lightweight structural nodes for the automotive industry. Prototypes are prepared using surface smoothing and mesh logic operations [80]. Two manufacturing technologies are considered. The direct manufacturing is done using the Laser Metal Powder Bed Fusion (LM-PBF) technology. Second, indirect, approach uses the Laser Polymer Powder Bed Fusion (LP-PBF) technology to create the initial model which is used, in turn, to create a mold used for the manufacturing using an Injection Molding (IM). Both methods have different limitations, but the shared feature is limitation of the minimal element diameter. For both methods the cause is different - in the LM-PBF thermal stress may lead to deformation of small elements, in the IM liquid metal flow properties enforce minimal element sizes. Preparing the optimization output for manufacturing is time consuming and tedious operation, and when elements are small there were cases, when technicians decided to remove small parts instead enlarging them to meet manufacturing requirements. Proposed workflow should aim to reduce human effort required to perform such tasks. To sum up, main assumption underlying the workflow are following:

- Focus on the Additive Manufacturing preparation process. Family of AM approaches share common, well identified limitations. Minimal element size requirement is the one that's the most pronounced during the preparation of structural optimization results. Chip removal manufacturing approaches require different processing and are not part of this research.
- Reduction of human effort required for the model preparation. Commonly used mesh smoothing, despite benefits such as preservation of complex structure of the input object, makes any change in mesh difficult. Standard industrial approach using the parametric model in a CAD system saves resources and allows better quality control of the final product.
- Automation of manual tasks. When limitations are known it is possible to apply them automatically on the parametric model. Ideally the structural optimization software itself should produce the result considering limitations of the manufacturing process. In practice this task is part of the model preparation step.
- Additive nature of mesh processing. Preparation of a mesh for the manufacturing process consists mostly of adding volume to the mesh. Elements are thickened or supporting structures are added to decrease the amount of supports required. One exception to this observation is reshaping the top surface of holes in PBF techniques

into 'drop' shape to remove need for supports. In general operations decreasing mesh volume degrade mechanical properties of the element and should be avoided hence are disregarded in this research.

Considering both specific of the researched, truss based, parametric models and practical considerations learned from prior experience in the BioniAMoto project hybrid workflow is proposed. Figure 8.1 depicts proposed workflow. Optimized element is prepared in the CAD system and the structural optimization step is performed as usual. Next step is extraction of the area excluded from the optimization. Mesh binary operation is performed as described in the Chapter 7. As the input data contains all information required to perform this task it can be fully automated. After the extraction is done, the process forks into two independent paths. First one is the classical approach – the mesh smoothing, which can be performed automatically, based on pre-defined settings based on the specifics of both the optimization process and desired manufacturing process. Second path uses the researched parametric model recognition which extracts the CAD model from the input mesh. Change introduced to this process is an automated step after the refinement. Minimal element size constraints are applied to the extracted parametric model. This path also includes manual model validation in the CAD system. In this step the operator can apply additional changes or correct the parametric model if it finds it needed. The workflow assumes the additive nature of such changes. The CAD model, at the end of this fork, is converted back into a finite element mesh. Meshes from both paths are fused together and then fused with areas excluded from the optimization. The final mesh is evaluated, any additional changes require going back to the Manual Model Validation step and redoing changes. The result mesh contains both manufacturing method restrictions applied to the extracted parametric model and fine details of the smoother mesh, not captured by the feature recognition system such as complex geometries around mesh nodes. The new idea, introduced in the proposed approach, automates the majority of steps required of the mesh preparation workflow. Synergistic application of two completely different approaches allows to retain complex mesh while allowing easy additive changes to the mesh using a CAD system.

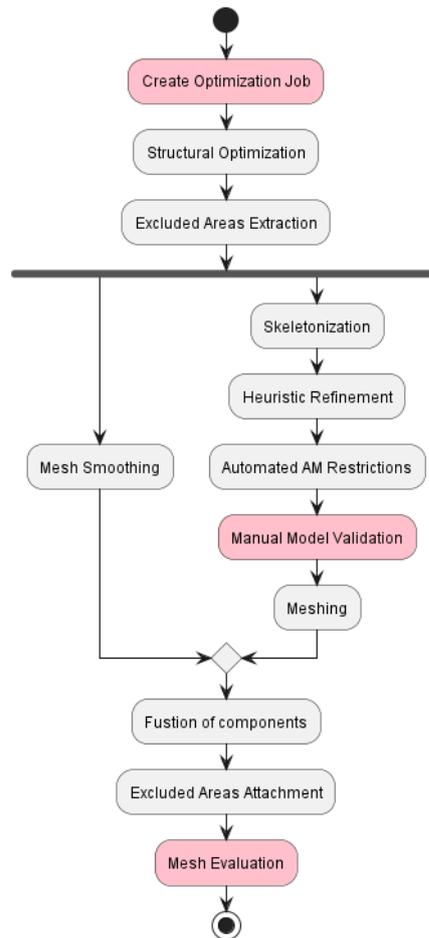


FIGURE 8.1. Proposed hybrid workflow integrating structural optimization process with additive manufacturing using both mesh smoothing and feature recognition approach. Highlighted manual steps of the workflow.

# Chapter 9

## Results

This chapter consists of four sections, each presents results of research described in previous chapters. The first section presents changes applied to the mesh generator. Second section shows the Hough Transformation algorithm applied to both test data created in a CAD software and the real samples from the biomimetic structural optimization system. Third section presents a parametric model generated by the skeletonization-based approach for test cases analyzed in the available literature and for industrial application from the BioniAMoto project. In the Fourth section the proposed workflow is applied to the structural optimization output.

## 9.1 Changes in the Mesh Generator Tool

Two directions of changes in the mesh generator were selected. The first improvement was an implementation of the mesh correction step to remove non-manifold elements by adding additional volume to the mesh. The second change was the extension of structural elements to allow more complex shapes. Algorithm for correction of non-manifold mesh elements was implemented and additional volume elements were added to places where manifold edges are detected. Mesh generator was extended with the support of a partial structural element used for the mesh building process.

Rat bone micro-CT, same used for an evaluation of the mesh generator, was used to test the correction algorithm. Mesh analysis with the FreeCAD Mesh Evaluator was performed, results are shown in Figure 9.1. The detail of a corrected mesh with non-manifold issue is shown in Figure 9.2. Analysis of the mesh generated with the corrected generator indicates no issues which allows further processing of the mesh with appropriate algorithms. The increase of the structural elements variety in the mesh generator greatly increased the quality of the mesh. Increased source data mapping capabilities are shown in Figure 9.3 and smoother surface of an output mesh is shown in Figure 9.4. Moreover, Laplacian Smoothing applied to data of higher quality now produces smooth meshes, unlike applied to the original mesh (Figure 9.5).

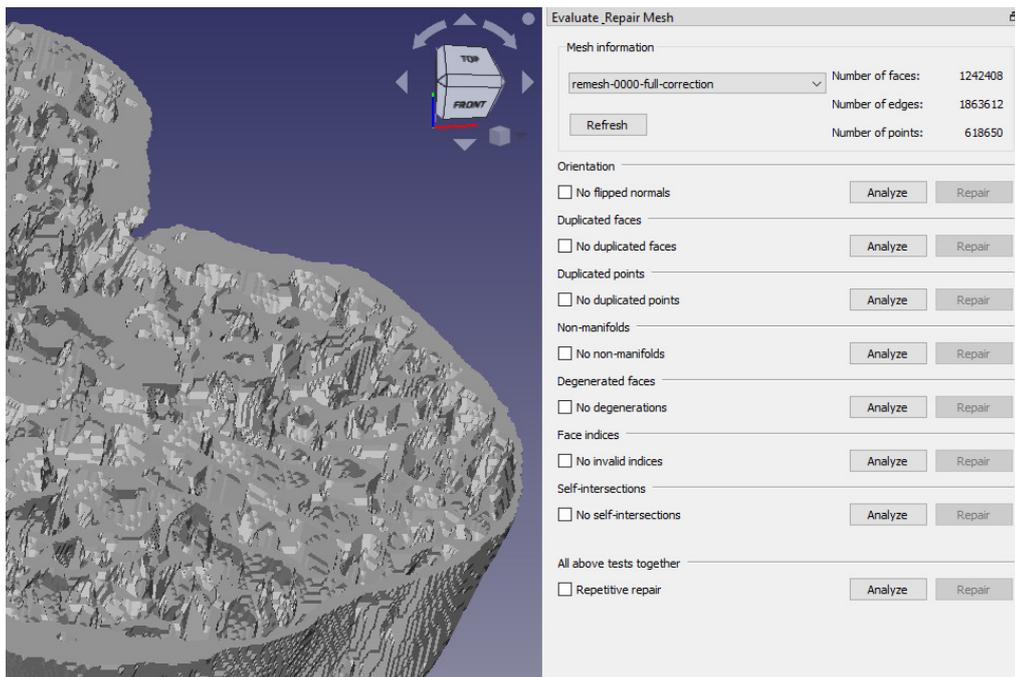


FIGURE 9.1. The result of analysis with FreeCAD Mesh Evaluator of the mesh generated for micro-CT of rat bone. Analysis indicated no mesh issues.

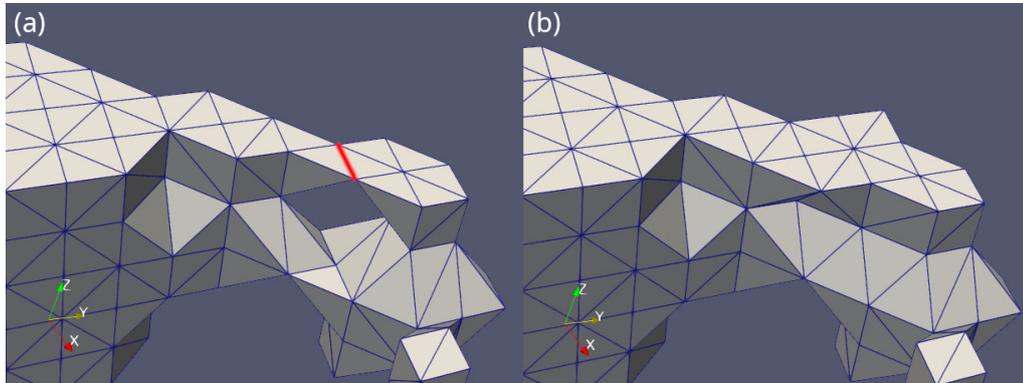


FIGURE 9.2. Showcase of the non-manifold repair algorithm on a mesh detail. (a) The original mesh, the non-manifold edge highlighted in red. (b) The corrected mesh. Additional volume in the area of the non-manifold edge was added.

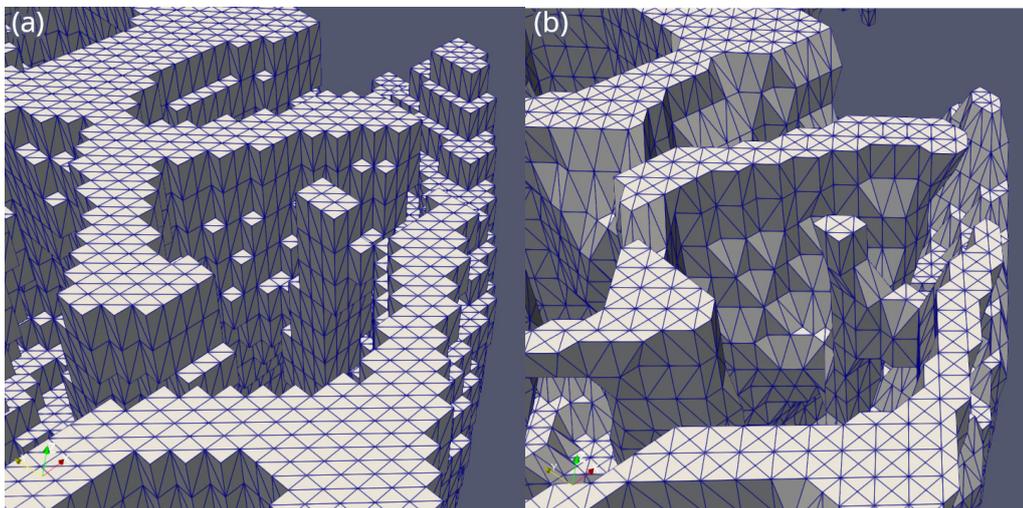


FIGURE 9.3. Detail of a mesh generated from rat bone micro-CT showing the quality of input data mapping. (a) Mesh created with the original mesh generator. (b) Mesh created with the new mesh generator. Addition of new structural elements enables better mapping of the input data into the output mesh.

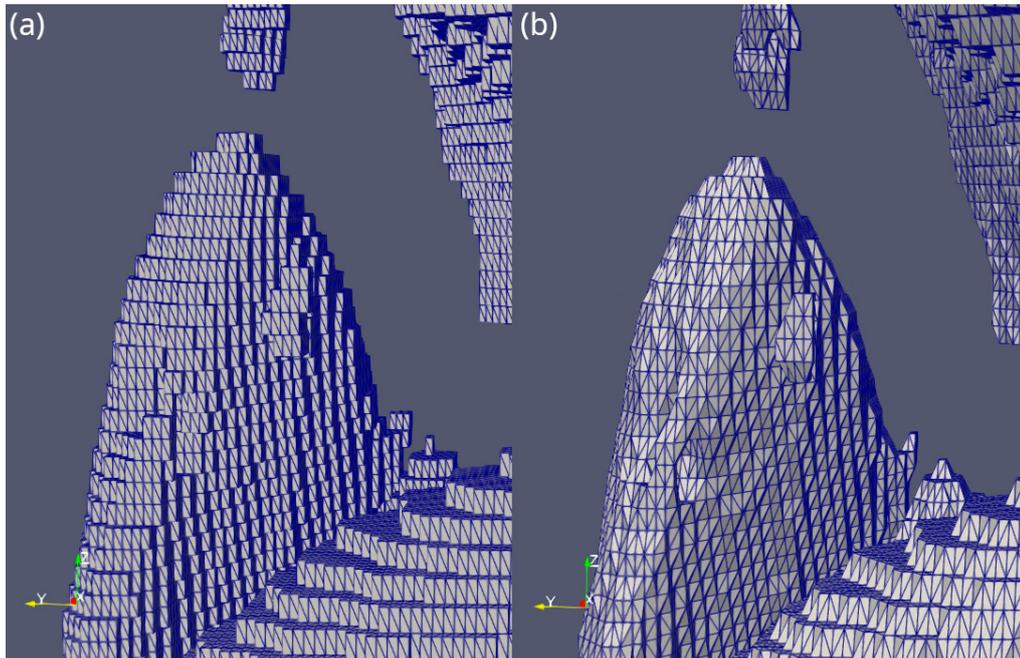


FIGURE 9.4. Detail of a mesh generated from rat bone micro-CT showing the surface quality of a mesh. (a) Mesh created with the original mesh generator. (b) Mesh created with the new mesh generator. Addition of new structural elements allows better surface mapping.

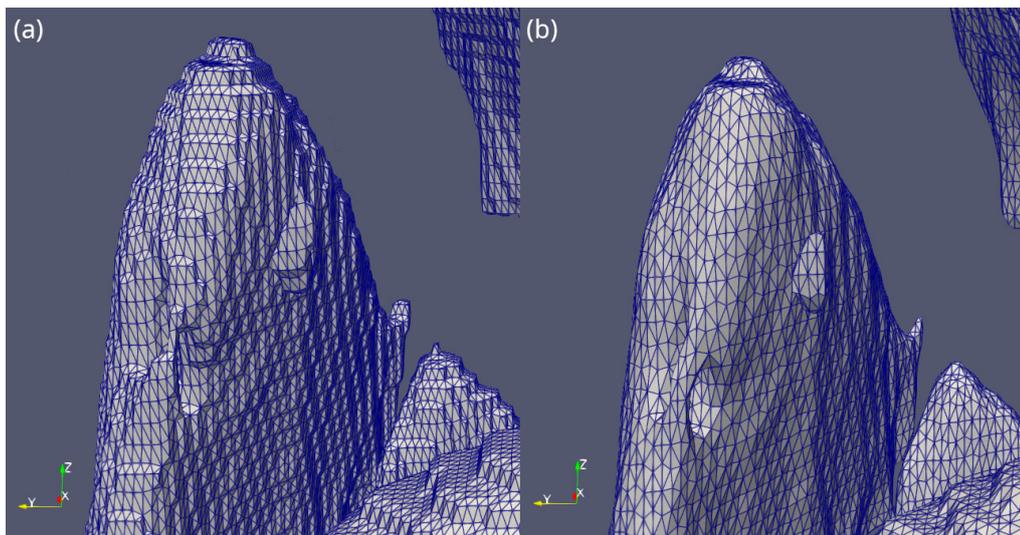


FIGURE 9.5. Mesh detail, same as in Figure 9.4 after 100 iterations of the Laplacian Smoothing algorithm. (a) Mesh created with the original mesh generator. (b) Mesh created with the new mesh generator. Smoother surface of the input data results helps to achieve better results after smoothing.

## 9.2 3D Shape Recognition using Hough Transform

### 9.2.1 Case Study 1—Single Capped Cylinder

First case study was a feature recognition run for a single cylinder with rounded sides (capped) created using FreeCAD software 9.6. As only the cylinder surface is recognised by the algorithm, the used shape will be called cylinder in the following text. Cylinder radius was 10mm, height 40mm. Caps on both ends added 2.5 from each side. Axis direction was vector  $x = 1, y = \sqrt{2}, z = 1$ . This test had a dual purpose. First goal was to validate the proper algorithm operation. It was based on the assumption, that parametric model converted to triangulated mesh, generated by available algorithms, conforms to the main observation underling used algorithm - direction vectors of normals of side faces of the cylinder form a circle which is an intersection of the Gaussian sphere and the plane normal to the cylinder axis. Second was to confirm that the algorithm properly operates using the mesh generated by the structural optimization mesh generator. Results obtained for such mesh were compared with results obtained in the first part of a test.

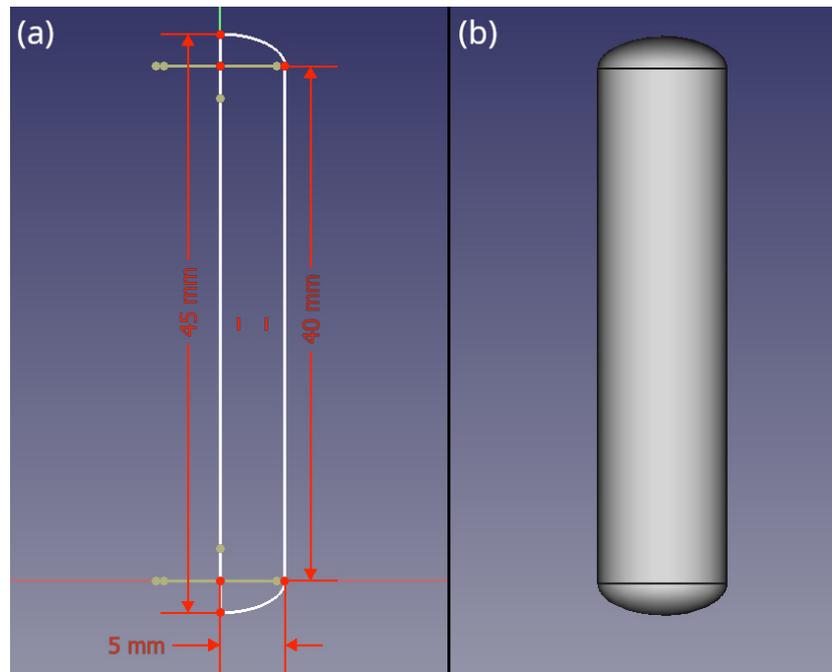


FIGURE 9.6. Parametric model of the element used for the first test. Sketch on the left was extruded by rotation around the vertical axis to obtain the element on the right.

First test was run for the parametric model directly exported from FreeCAD as STL mesh using default settings. Mesh is visible in Figure 9.7. This mesh contains faces of varied sizes - the side of cylinders contains big elements while ends are built of a large number of faces. Each face contributes the same value to the accumulator because the algorithm was designed to work with structural optimization output mesh which contains faces of similar sizes so no correction was considered. To run tests on mesh that has faces on uniform size alternative meshing of the parametric model was used. Two meshing algorithms, available in FreeCAD, were used. First was generated using the Mefisto Mesher [81] algorithm and the second was generated using the gmsh MeshAdapt algorithm [66]. Both algorithms

take element size as an input. For Mefisto Mesher it is a single value which was set to 0.4mm. MeshAdapt takes min and max values which were set, respectively, to 0.3mm and 0.4mm. Outputs of mesh generators are shown in Figure 9.8. It is worth noticing MeshAdapt generated mesh with irregular surface and elements of roughly equal sizes. Mefisto Mesher, on the other hand, generated regular mesh, but with element size way smaller than requested - equilateral triangles on a side surface of the mesh have side length of 0.3mm. Elements near the ends of the cylinder have a high aspect ratio.

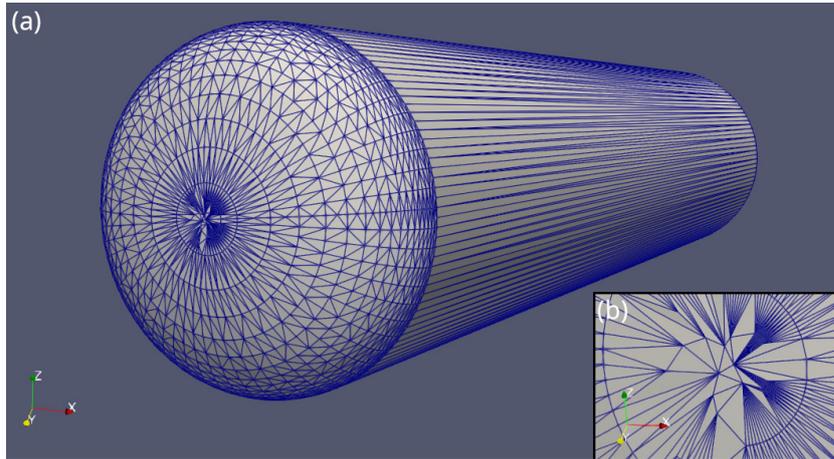


FIGURE 9.7. (a) Mesh generated for the parametric model of the test cylinder using the default export settings in FreeCAD software. The side surface consists of big, high aspect ratio elements while the ends contain very detailed, small elements. (b) Detail of the cylinder end. Faces of varied size, with high aspect ratio, are visible.

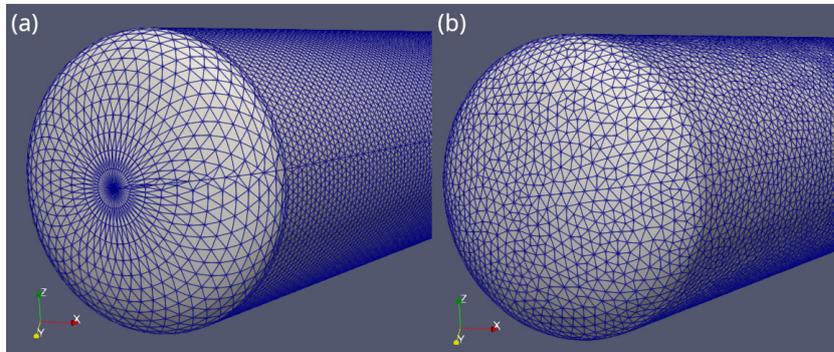


FIGURE 9.8. Meshes for the parametric cylinder generated using (a) Mefisto Mesher and (b) MeshAdapt algorithms. Both meshes were generated for element size = 0.4mm.

The mesh created using the mesh generator from the structural optimization system was generated using an STL file exported from FreeCAD software. Mesh was sliced into 100 slices. The total element height was 30mm, so size of face was between  $\frac{30}{100} = 0.3mm$  and  $\sqrt{2}\frac{30}{100} = 4.24mm$ . This gives the same element sizes as in two meshes generated earlier. Mesh generated this way consists of faces that have a limited set of possible spatial alignments. This means mesh faces do not have the required property of normals pointing towards the cylinder axis. To fix mesh smoothing was applied, which is standard mesh post processing technique [82]. Laplacian smoothing, available in the Paraview software, using default settings and 200 iterations was applied, as shown in Figure 9.9. The used algorithm has a major drawback - causes volume shrink, but is readily available and fast

so it was considered sufficient for the initial testing phase.

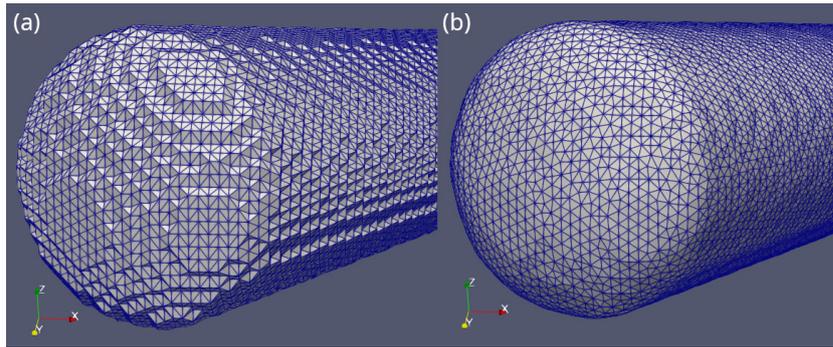


FIGURE 9.9. Meshes for the parametric cylinder generated using the mesh generator from the structural optimization system. The parametric model exported as an STL file was used as an input. (a) Direct output. (b) Output with Laplacian Smoothing algorithm applied.

Meshes generated with selected algorithms were used as input of the feature recognition algorithm. Parameters of extracted features are in Table 9.1 and differences between source data and extracted features are in Table 9.2. As the first stage of recognition is critical and any error introduced there is further propagated in subsequent stages of the algorithm, Hough accumulators generated at this step are useful for visual inspection of algorithm's behavior – Figure 9.10. For completeness the accumulator calculated for mesh directly obtained from structural optimizer mesh generator is also shown. It is visible how a limited set of face orientation makes application of feature recognition impossible without data pre-processing. Results for the first algorithm, Mefisto Mesher, are very interesting - magnified detail of Hough accumulator for the direction estimation step is on Figure 9.11. Mesh generated by this algorithm doesn't meet the assumption underlying feature recognition algorithm - faces on a side of the cylinder are slightly slanted and circles in Hough accumulator 'miss' the right point. Because the maximum value from the accumulator is not a single point, but also includes the neighborhood, this phenomenon is compensated and the axis is correctly estimated as a center of the circle. This shows sensitivity of the feature recognition algorithm to input data. On the other hand this particular problem can be solved by Laplacian smoothing to obtain the perfect mesh alignment. For the MeshAdapt algorithm minimal error was obtained, error is slightly higher than for the Mephisto Mesher algorithm. In the Hough accumulator all values are concentrated around the right value. Results for the smoothed mesh from the structural optimizer were the worst but still in acceptable range. Radius detected for this mesh is smaller than actual which can be attributed to volume shrink inherent to the used smoothing method.

TABLE 9.1. Comparison of the source cylinder parameters with the results of the algorithm for three different mesh generators.

	<b>A</b>	<b>B</b>	<b>Radius</b>
Source	(0, 0, 0)	(20, 28.23, 20)	5
Mefisto Mesher	(-0.181, -0.157, -0.069)	(20.07, 28.28, 19.94)	5
MeshAdapt	(-0.180, -0.156, -0.068)	(20.19, 28.45, 20.07)	5
Optimizer	(-0.387, -0.26, -0.245)	(20.16, 28.22, 20.17)	4.9375

TABLE 9.2. Proposed quality measures calculated for three input meshes.

	$\Delta A$	$DirErr(\mathbf{a})$	$\Delta B$	$DirErr(\mathbf{b})$	$AxisErr$
Mefisto Mesher	0.249	0.080	0.14	0.114	0.253
MeshAdapt	0.248	0.080	0.199	0.116	0.253
Optimizer	0.526	0.166	0.135	0.130	0.389

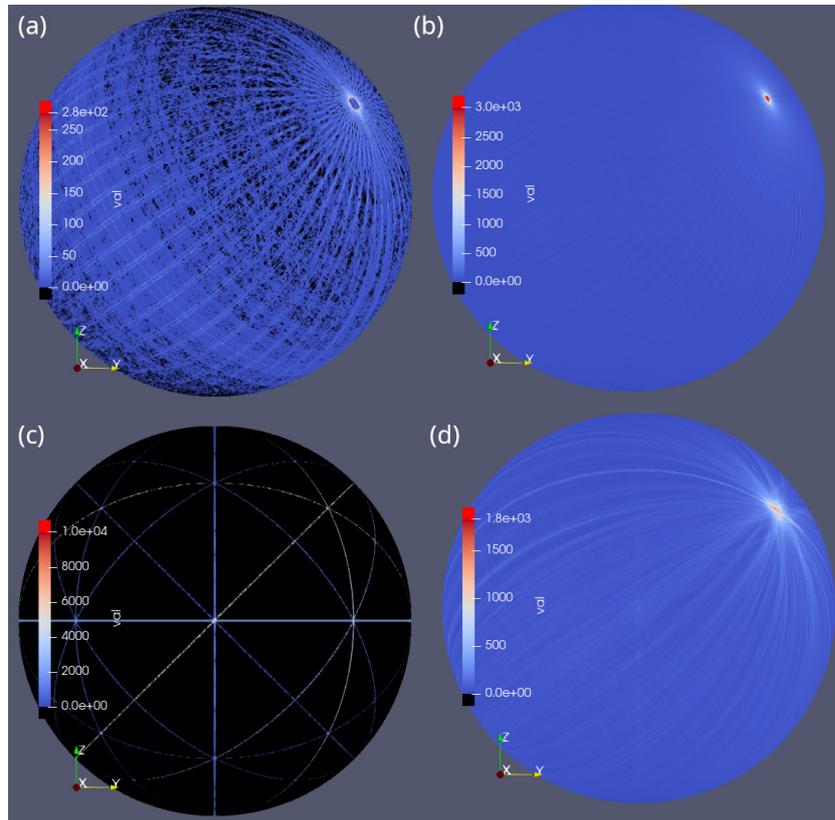


FIGURE 9.10. Hough space accumulators obtained for the first step of the algorithm - the direction estimation. (a) Mefisto Mesher. (b) MeshAdapt. (c) Structural Optimizer Mesh Generator. (d) Structural Optimizer Mesh Generator with Laplacian smoothing. Detection based on Structural Optimizer Mesh Generator output is impossible but smoothing mesh fixes this problem.

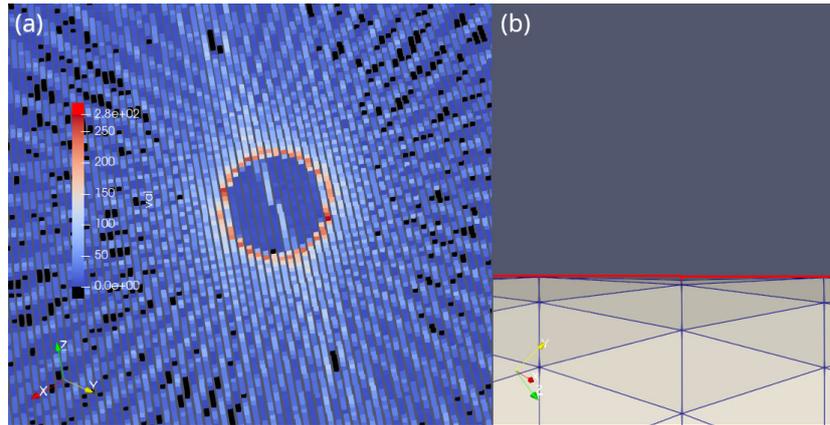


FIGURE 9.11. (a) Magnification of Hough space accumulator area corresponding to the cylinder axis obtained for the mesh generated with the Mephisto Mesher algorithm. Circular region indicates that the mesh generated with this algorithm contains some issue. (b) Magnification of the surface of the cylinder mesh generated with the Mephisto Mesher algorithm. Red line is the surface of the parametric cylinder. Visual inspection indicates surface faces are slightly angled which explains the obtained pattern in the accumulator.

### 9.2.2 Case Study 2—Complex Parametric Object

Second test was run on parametric object design in a cad system to test algorithm features not tested in the first series of tests. Object used is shown in a Figure 9.12. Object was designed in the CAD system as a benchmark for the algorithm. It consists of 4 parts. It contains 3 cylinders based on an extruded circle and one based on an extruded ellipse with major axis  $a = 4.8mm$  and eccentricity  $\varepsilon = 0.85$ . All features had rounded ends, as described before, to avoid algorithm issues with planes in the input data. Ellipse was added to test the algorithm's behavior for non-circular features. Two cylinders had similar axes, angle between them is  $3^\circ$ . One was much bigger ( $R = 5mm, h = 40mm$ ) than the other ( $R = 3mm, h = 25mm$ ). It is possible for a bigger object to “shadow” a smaller one during the Orientation Estimation step, making detection using a single pass of the Orientation Estimation step impossible. Successful detection therefore justifies this difference in implementation compared to the [62]. Previous tests demonstrated adequate behavior for those meshes, there was no need to run more tests on the perfect data obtained by meshing the parametric model. The mesh generated by the structural optimization system was used with three different levels of smoothing applied as a mesh preprocessing step. The Laplacian smoothing algorithm was applied using 100, 200 and 500 iterations. Source mesh and smoothed meshes are in Figure 9.13.

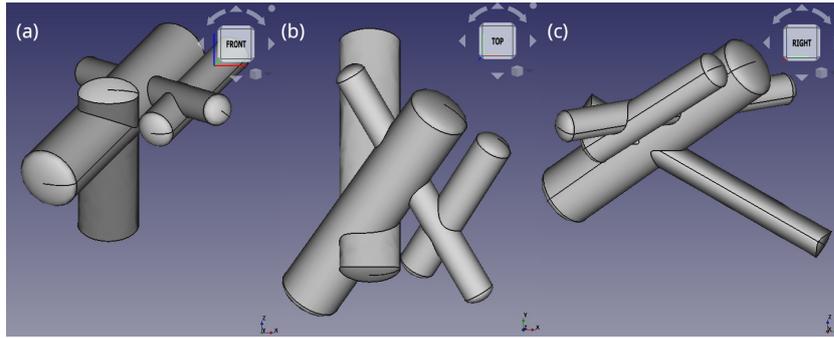


FIGURE 9.12. Parametric object used in Case Study 2. The shape was designed to test the feature recognition algorithm on a more complex data than a simple cylinder before testing on a real data. Big and small cylinders are not collinear, there is  $3^\circ$  of a difference between axes.

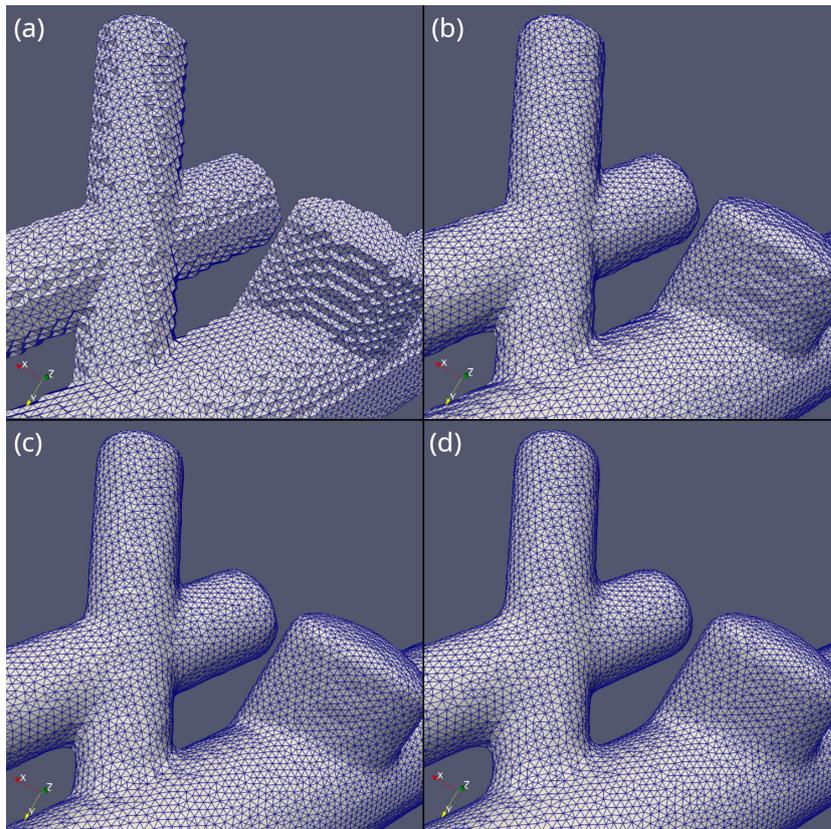


FIGURE 9.13. A detail of the input mesh. (a) Directly from the structural optimizer mesh generator. (b, c, d) Post-processed using the Laplacian smoothing algorithm with 100, 200 and 500 iterations respectively. As a surface smoothness increases fine details are lost.

For each mesh feature recognition algorithm was run and recognition accuracy was computed. Parametric model generated for each mesh is shown in Figure 9.14 and accuracy measures are in Tables 9.3, 9.3, 9.5. Algorithm detects cylinders with circular cross-section without issues. Two cylinders with closely aligned axes are estimated correctly. Figure 9.15 and Figure 9.16 show progress of the algorithm. Initial Hough space accumulator shows only single response at area corresponding to two cylinder axes due to noise in the input data. In each test elliptical elements were estimated incorrectly. Axis was estimated correctly, but the circle detection algorithm is not able to properly recognise ellipses. Elliptical element for the mesh with 100 iterations of the Laplacian Smoothing was estimated as a cylinder with the radius of  $4.625mm$  which is close to the value of the ellipse major axis -  $4.8mm$ .

As the smoothing increases the elliptical element estimation accuracy decreases. Algorithm detects one side of an ellipse as part of a big cylinder. Effect of the pre-processing of finite elements mesh is clearly visible in results. As more smoothing is applied, axes estimation accuracy increases but volume shrinkage becomes prominent as radius estimation accuracy decreases - radiuses values are underestimated. Nevertheless for the next tests a smoothing value of 500 is used. The right axis estimation is considered more important for the feature recognition as incorrect radius can be easily corrected in the parametric model in CAD software.

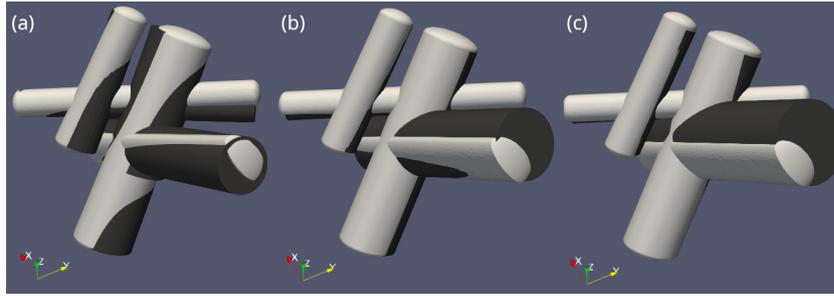


FIGURE 9.14. Results of the feature detection for the input mesh processed with the Laplacian smoothing using (a) 100 iterations, (b) 200 iterations, (b) 500 iterations. Input parametric model in light gray, reconstructed parametric model in dark gray. The biggest feature in (a) consists of two elements. The horizontal feature with an elliptical cross-section in both (b) and (c) was detected as a part of a bigger cylinder instead.

TABLE 9.3. Accuracy measures calculated for the Case Study 2 results for the mesh processed with 100 iterations of the Laplacian Smoothing algorithm

	$\Delta A$	$DirErr(\mathbf{a})$	$\Delta B$	$DirErr(\mathbf{b})$	$AxisErr$	$R$	$\Delta R$
Part 1	0.99	0.79	1.53	1.50	3.17	4.938	0.063
Part 2	2.74	2.47	1.65	1.24	4.95	4.625	—
Part 3	1.88	1.63	0.97	0.64	4.99	2.938	0.063
Part 4	0.23	0.18	3.14	3.13	4.68	2.375	0.125

TABLE 9.4. Accuracy measures calculated for the Case Study 2 results for the mesh processed with 200 iterations of the Laplacian Smoothing algorithm.

	$\Delta A$	$DirErr(\mathbf{a})$	$\Delta B$	$DirErr(\mathbf{b})$	$AxisErr$	$R$	$\Delta R$
Part 1	0.49	0.31	0.34	0.24	0.25	4.875	0.125
Part 2	3.22	3.00	3.88	3.78	1.06	5.936	—
Part 3	0.95	0.31	0.92	0.26	0.35	2.875	0.125
Part 4	0.30	0.30	0.32	0.21	0.25	2.375	0.125

TABLE 9.5. Accuracy measures calculated for the Case Study 2 results for the mesh processed with 500 iterations of the Laplacian Smoothing algorithm. As the input mesh smoothing increases so does the axis detection accuracy, but volume shrinkage is noticeable as radii are underestimated. Element 2, with elliptical cross-section, is incorrectly estimated and error increases as more smoothing is applied.

	$\Delta A$	$DirErr(\mathbf{a})$	$\Delta B$	$DirErr(\mathbf{b})$	$AxisErr$	$R$	$\Delta R$
Part 1	0.28	0.26	0.60	0.37	0.57	4.88	0.125
Part 2	4.79	4.70	4.85	4.82	0.19	7.06	—
Part 3	0.69	0.68	1.27	0.26	0.42	2.88	0.125
Part 4	0.35	0.30	0.20	0.12	0.26	2.31	0.1875

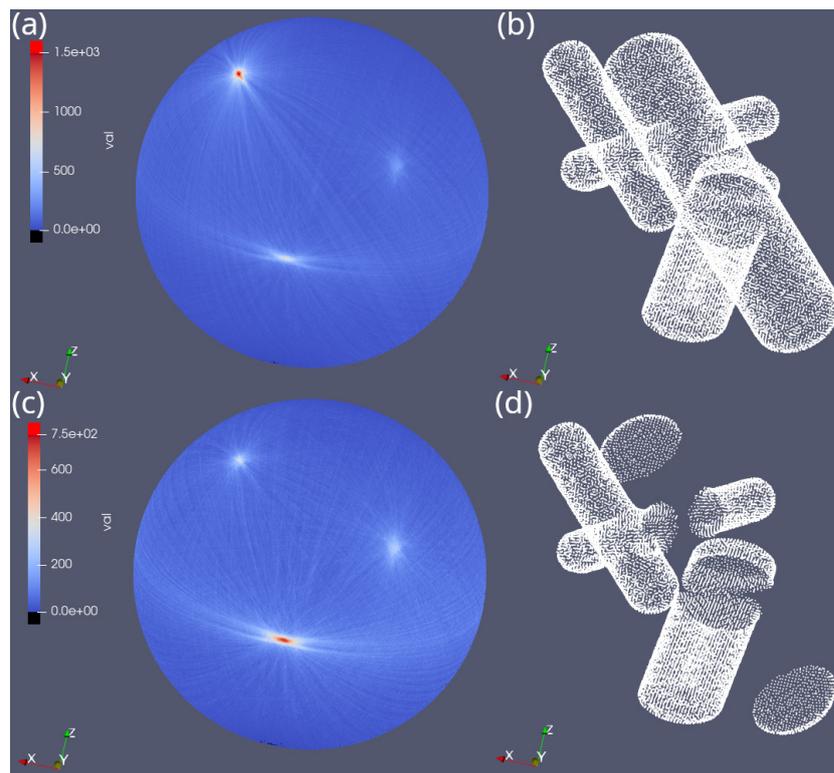


FIGURE 9.15. Progress of feature detection algorithm for the mesh after 500 iterations of the Laplacian Smoothing. The hough space accumulator for the initial state (a) and after the first iteration (c). Initial mesh as a points cloud (b) and mesh as a points cloud after the first iteration (d). Removal of the feature detected during the first step uncovered another feature, previously hidden under votes generated by the recognised feature.

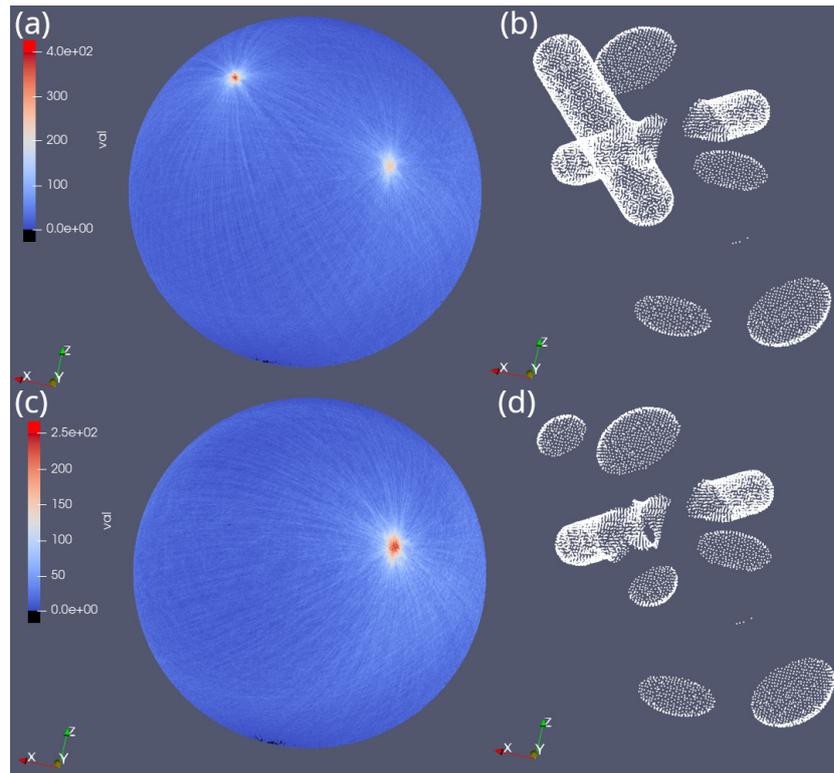


FIGURE 9.16. Progress of the feature detection algorithm for the mesh after 500 iterations of the Laplacian Smoothing. The hough space accumulator for the state before the third iteration (a) and after the third iteration (c). Mesh before the third iteration as a points cloud (b) and mesh as a points cloud after the third iteration (d).

### 9.2.3 Case Study 3—Optimized Cantilever Beams

The third series of tests has been run using the actual output of the structural optimization system. Four different results of cantilever beam optimization were selected. All data sets were obtained for the steel bar of size  $1000\text{mm} \times 20\text{mm} \times 20\text{mm}$ , attached on one end and loaded on the other. Optimizations were performed with the biomimetic structural optimizer for different load conditions and domain sizes during optimizer tests. Meshes for tests were selected to include meshes with simple topologies such as 4 straight bars (Mesh 1), 3 bars, one splitted in  $\frac{1}{3}$  of the length (Mesh 2), complex topology with five support points (Mesh 3). Last mesh is the classical cantilever beam optimization in domain limited to two dimensions (Mesh 4). All meshes were pre-processed using the Laplacian smoothing algorithm. All test meshes are shown in Figure 9.17.

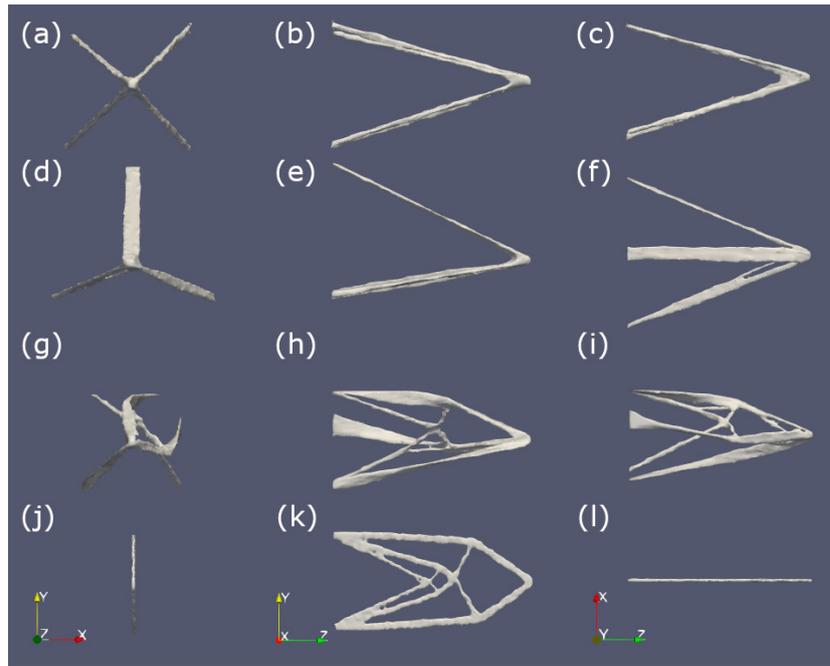


FIGURE 9.17. Test data used for feature recognition in Test Case 3. All meshes were obtained using the structural optimization algorithm for the cantilever beam under different input conditions. All meshes were pre-processed using the Laplacian smoothing algorithm. (a, b, c): Test mesh 1. (d, e, f): Test mesh 2. (g, h, i): Test mesh 3. (j, k, j): Test mesh 4. (a, d, g, j): View of test meshes along the Z axis. (b, e, h, k): View of test meshes along the X axis. (c, f, i, l): View of test meshes along the Y axis.

### Case Study 3—Mesh 1

The first mesh is the optimization of the cantilever beam for multiple load scenarios. Two forces, one along X and the other along Y axes were considered. The optimization domain was big enough to allow formation of a simple solution without the need of internal microstructure, commonly seen in solutions with the constrained optimization domain. This mesh was selected as the first test because the solution is very simple – four bars arranged in the letter X when looking along the Z axis. Each of the bars appeared to be straight and roughly cylindrical. The feature recognition algorithm was able to properly detect mesh features. Parametric model imported into the CAD system is shown in Figure 9.18. This particular example shows the importance of the feature recognition algorithm as a tool for post processing of structural optimization output. Solution that is considered to be symmetric, in fact, is not. Two bars are visibly thicker than the other pair. Moreover, as visible in the panel (c) of Figure 9.18, bars in the optimization solution are not straight, but slightly bent. More research is required to compare the implied symmetric solution with the actually recognised asymmetric.

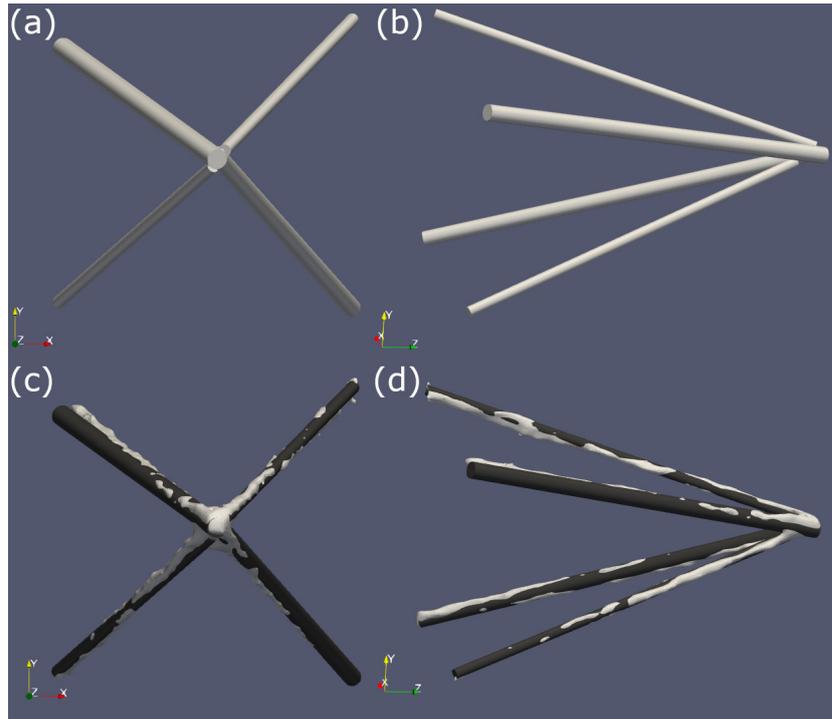


FIGURE 9.18. (a, b): Parametric model obtained for the Mesh 1. (c, d): Input mesh (white) overlapping reconstructed parametric model (black). Although the input mesh looks symmetric to the naked eye, the feature recognition process uncovered differences in the thickness between elements.

### Case Study 3—Mesh 2

The second analyzed mesh was obtained using structural optimization with input conditions similar to the Mesh 1, but input force directions were randomly altered each step by small angle resulting in overall sturdier structure, able to sustain loads slightly different than the one load used for the structural optimization. In this particular run of the structural optimization algorithm, the resulting mesh was shaped in the letter Y along the Z axis. Moreover this test mesh was selected because it exhibits two interesting features, compared to previous one - one of “legs” is flat and other splits into two in approximately  $\frac{1}{3}$  of total length. Feature recognition results are visible in Figure 9.19. The first detected part is the cylinder visible in the top-right part of the panel (d). It is the only part properly detected. Other three features were detected with the right axis directions, but the radius detection failed. Flat surfaces on each of the mesh “legs” were detected as arcs of big circles causing big errors in resulting parametric models. Sample output of the radius and position estimation step is shown in Figure 9.20. The algorithm searches for the perfect circle in the data set to estimate radius and position of the cylinder and it finds a big radius gets the highest score in a noisy input data. On the other hand orientations were estimated correctly, so the output of the algorithm is still usable for the semi-automatic reconstruction in a CAD software. Element containing split wasn’t properly recognised by the algorithm - incorrectly overestimated cylinder radius resulted in a cylinder covering the whole feature.

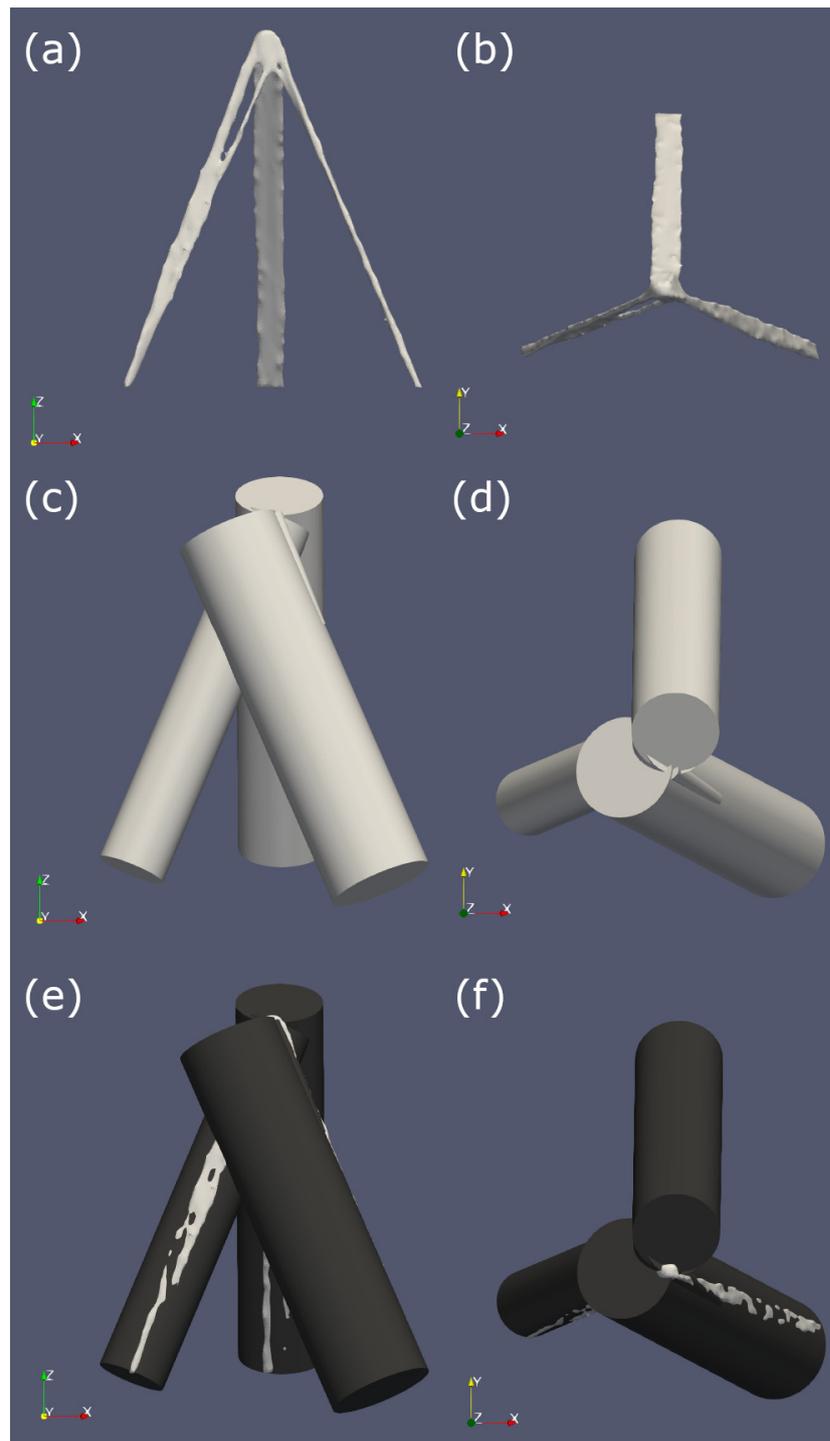


FIGURE 9.19. Feature recognition results for the Mesh 2. (a, b) Input mesh. ©, (d) Parametric model. (e, f) Input mesh (white) and parametric model (black) showing feature recognition details. The first recognised feature was the small cylinder on the top-right part of (d) - both direction and radius were recognised correctly. Other three features have right axis direction but radius detection algorithm failed. Slightly curved surfaces of the input mesh were detected as parts of big circles.

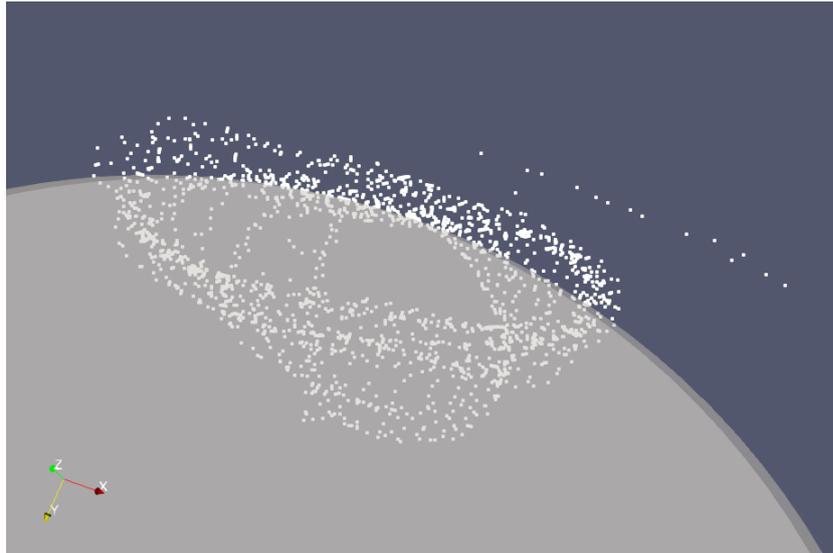


FIGURE 9.20. Mesh surface points projected on the plane normal to the detected cylinder axis (white dots) and the cylinder estimated using this data (gray) for the third iteration of the algorithm running for the Mesh 2.

### Case Study 3—Mesh 3

Third analyzed mesh is an optimization result obtained for multiple load scenarios with limited domain. Small numerical error during the initial mesh discretization step caused high asymmetry of the output mesh. Mesh consists of both cylindrical elements and highly flattened parts. Feature detection results are in Figure 9.21. It is clearly visible that the algorithm again fails to properly detect elements with a non-circular cross section. Properly detected elements were extracted from the parametric model and are visible in Figure 9.22.

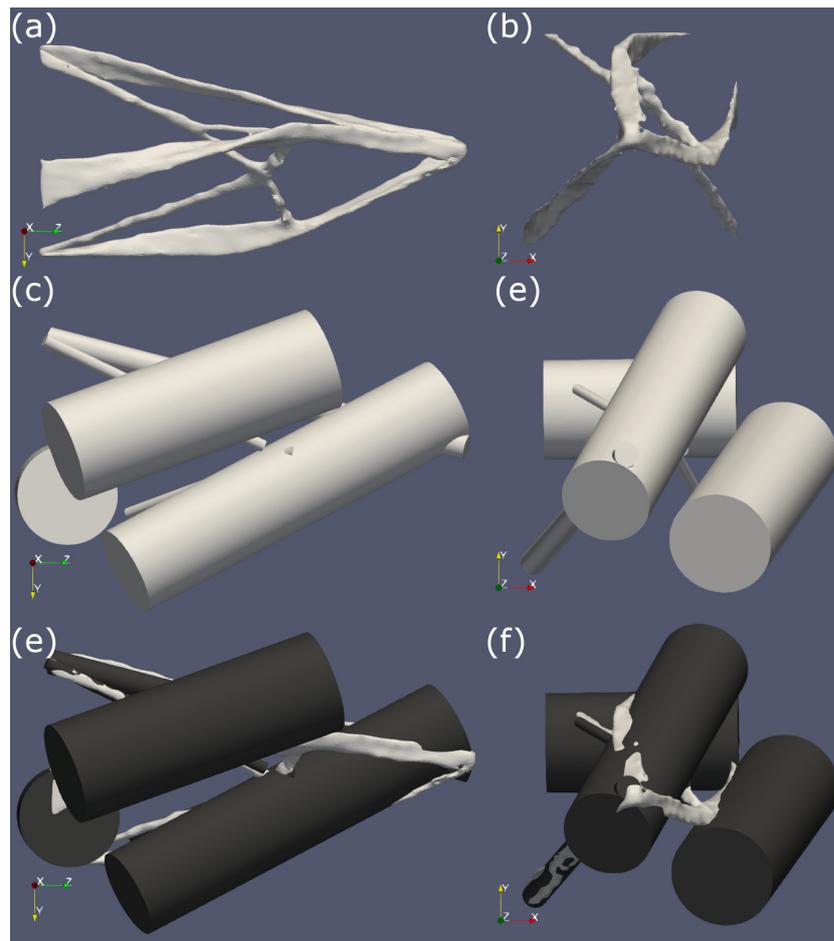


FIGURE 9.21. Feature recognition results for the Mesh 3. (a, b) Input mesh. (c, d) Parametric model. (e, f) Input mesh (white) and parametric model (black) showing feature recognition details. The same problem as with the Mesh 2 appeared, despite some features being recognised correctly. Note how the flat area on the left side of the mesh (a) was recognised as a cylinder with incorrect axis.

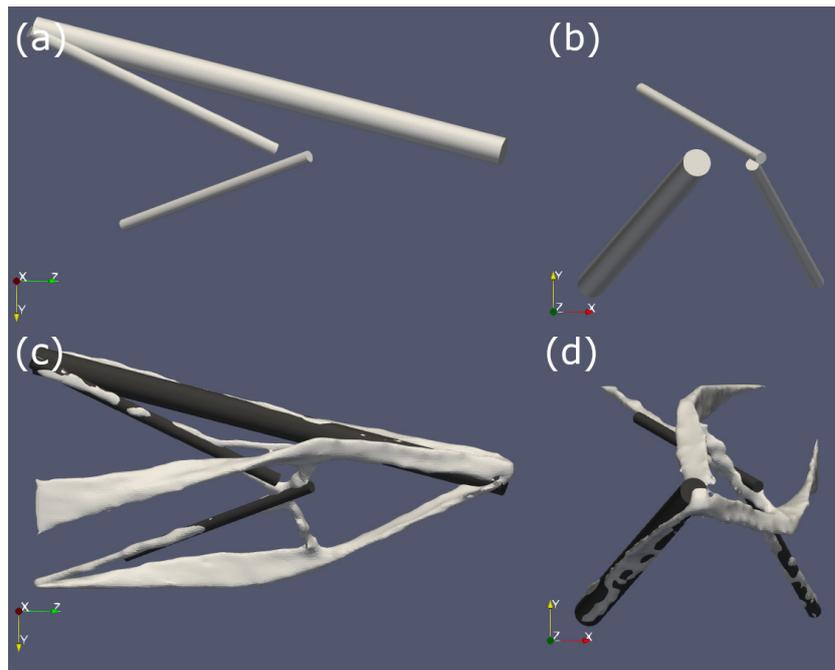


FIGURE 9.22. Feature recognition results for the Mesh 3 limited to correctly detected features. (a, b) Parametric model. (c, d) Input mesh (white) and parametric model limited to correctly recognised features (black).

### Case Study 3—Mesh 4

Fourth analyzed mesh is the classical example of 2D cantilever beam with single load. The algorithm completely fails to detect features, as seen in Figure 9.23. Flat areas interfere with detection of cylinders. Moreover further smoothing of the mesh is not possible because, as shown in Figure 9.24, it leads to high volume reduction and overall mesh quality degradation. In all previous case studies axis estimation was correct and subsequent steps were failing. As radius and position estimation works on the 2D projection along the estimated axis it would be easy to replace this step with the more sophisticated approach detecting other geometries. In this particular case axis estimation is incorrect.

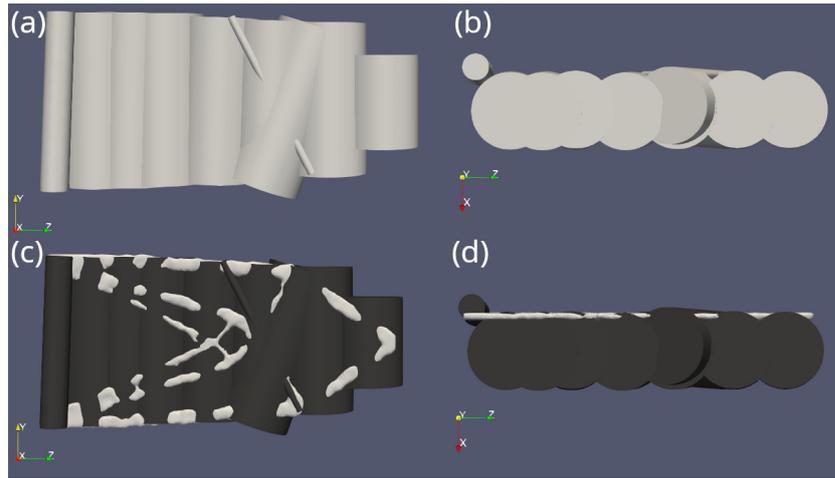


FIGURE 9.23. (a, b): Parametric model obtained for the Mesh 4. (c, d): Input mesh (white) overlapping reconstructed parametric model (black). Algorithm failed completely. Flat sides of the mesh interfered with the radius estimation algorithm leading to wrong results. Algorithm failed to extract any usable knowledge from the input data.

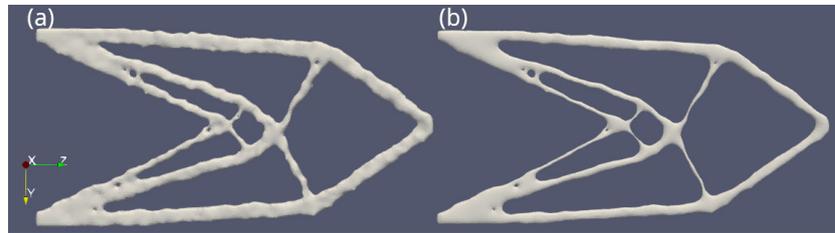


FIGURE 9.24. (a) The mesh used for the feature recognition, obtained from the structural optimization result mesh by applying 500 iterations of the Laplacian Smoothing algorithm. (b) The structural optimization result mesh after applying 2000 iterations of the Laplacian Smoothing algorithm. Increased smoothing increases curvature of flat areas, but distorts fine details and greatly decreases overall mesh volume.

An attempt to manually fix the mesh was made to test if removal of flat areas increases algorithm accuracy. The algorithm operates on a triangulated surface mesh, hence it is possible to manually remove triangles that cause issues for the algorithm. To remove flat areas the Clip data filter in ParaView software was used. This particular filter has the “crinkle clip” option, which causes the mesh triangles to be kept intact, instead of splitting them by the clip plane. Two clip filters, one for each side of the input mesh were used. The result mesh and the parametric model computed for the mesh are in Figure 9.25. Removal of flat areas not only didn’t help, but also reduced the amount of mesh data points below a level where the detection is possible.

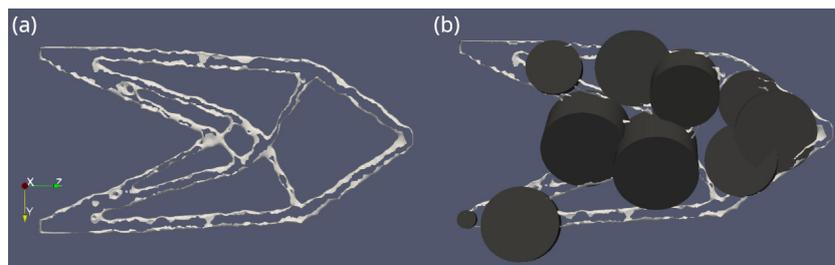


FIGURE 9.25. An attempt to increase the algorithm accuracy by manually removing flat sides from the input data. (a) Input mesh. (b) Input mesh (white) overlaid with the parametric model (black). After removal of mesh sides there are not enough data points to perform the feature recognition.

## 9.3 Automated, Skeletonization-Based Truss Recognition

### 9.3.1 Comparison with the Literature

Process results were tested using two structural optimization solutions for which analysis has been carried out in the past. First is the output of Bremicker's cantilever beam results using the homogenization method [75]. The parametric model for a 2D solution was provided by [6, 9, 75]. The obtained 2D result was extruded into a 3D mesh. The size of a bounding box of the mesh was  $230 \text{ mm} \times 144 \text{ mm} \times 18 \text{ mm}$ . The input mesh, skeletonized output and initial and refined solutions are visible in Figure 9.26. The obtained skeleton is noticeably higher quality than the one presented in the original work and allows capturing of the solution's topology. The second test is the cantilever beam provided in [6] and also analyzed in [9]. A similar but not exactly the same part was analyzed in [8, 24]. Again, the 2D mesh was extruded and the size of a bounding box of the mesh was  $600 \text{ mm} \times 162 \text{ mm} \times 25 \text{ mm}$ . Results are illustrated in Figure 9.27, demonstrating the proper operation of the feature recognition system. Visual inspection, as demonstrated in Figure 9.28, indicates the proper operation of the algorithm. The cited research did not perform stress analysis; hence, it was not performed for the obtained parametric models.

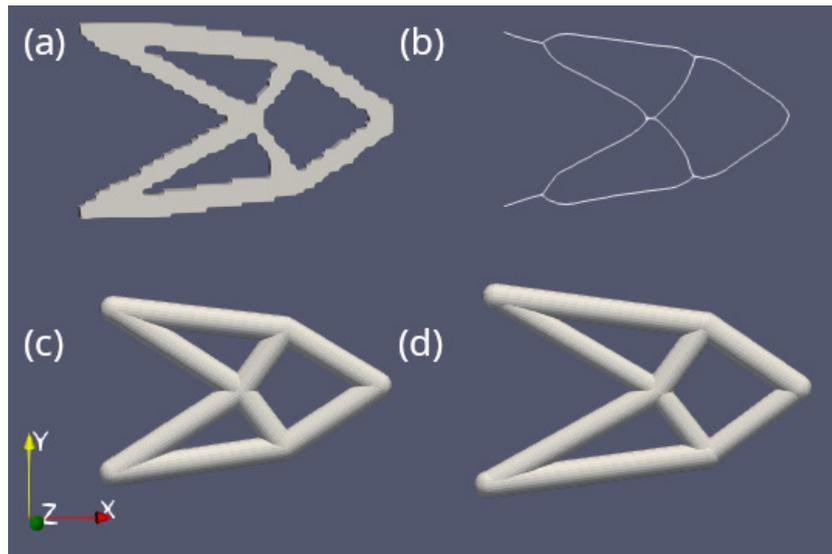


FIGURE 9.26. Bremicker's cantilever beam [75] processed by the feature recognition algorithm. (a) Structural optimization result. (b) Skeletonized mesh. (c) Initial parametric solution. (d) Final refined parametric solution.

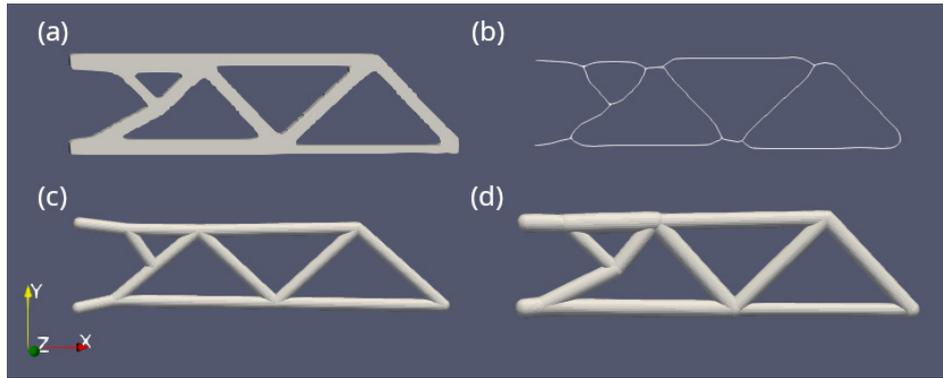


FIGURE 9.27. Gedig's cantilever beam [6] processed by the feature recognition algorithm. (a) Structural optimization result. (b) Skeletonized mesh. (c) Initial parametric solution. (d) Final refined parametric solution.

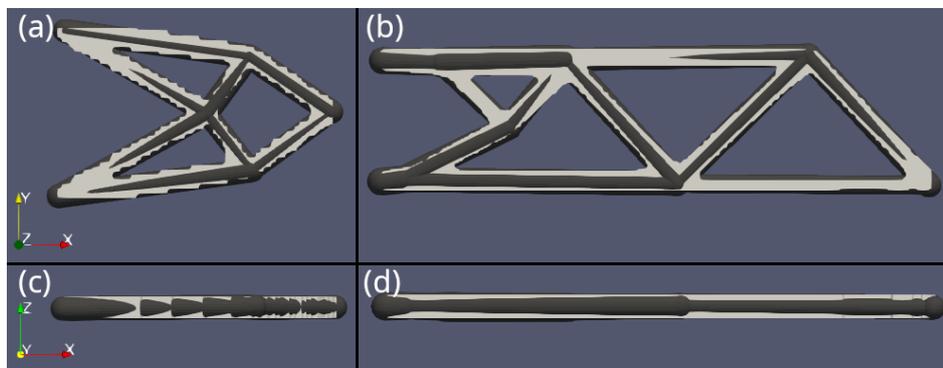


FIGURE 9.28. (a,c) Mesh of Bremicker's cantilever beam [75] (light) overlaid with the reconstructed parametric model (dark). (b,d) Mesh of Gedig's cantilever beam [6] (light) overlaid with the reconstructed parametric model (dark). Both shown cases visually demonstrate the quality of feature recognition results. In both cases, the reconstructed parametric solution matches the shape of the input.

### 9.3.2 Case Study 1—Cantilever Bar Bending

The first case is the optimization of loading the steel bar. This is a standard example of topology optimization and this problem has been solved analytically [83]. The structural optimization algorithm is used to obtain the input for the feature recognition algorithm, which was proved to provide the optimal solution [41]. It is important to note that the examples in the comparison with the literature were performed on extruded 2D optimization results. This case study is performed for a true 3D object with an optimization domain with one dimension restricted. The input for the optimization was created in CAD software using the following parts: optimization domain—box  $20 \text{ mm} \times 500 \text{ mm} \times 1000 \text{ mm}$ ; bar to be optimized—box  $20 \text{ mm} \times 20 \text{ mm} \times 1000 \text{ mm}$  centered inside domain; support area—box  $1000 \text{ mm} \times 1000 \text{ mm} \times 2 \text{ mm}$ ; stressed area—box  $20 \text{ mm} \times 20 \text{ mm} \times 2 \text{ mm}$ . All parts modeled in CAD are visible in Figure 9.29. The parts for use with the structural optimization system were exported as STL files. The material of the optimized model was steel with  $\nu = 0.28$ ,  $E = 200 \times 10^9 \text{ Pa}$ ,  $\rho = 7800 \text{ kg/m}^3$ . The yield strength for the material was assumed to be  $250 \times 10^6 \text{ Pa}$ . The bar was stressed with a force of  $40 \text{ kN}$  perpendicular to the axis. For this, a test factor of safety = 3 was assumed and the max allowed von Mises stress was set to  $80 \times 10^6 \text{ Pa}$ .

Structural optimization was run for 300 iterations. It is important to note that this case study has been prepared to compare results with examples widely used as a benchmark for structural optimization and the implementation of the results is not possible. A shear force of 40 kN is applied to the area of 20 mm  $\times$  20 mm and this alone would cause structural failure.

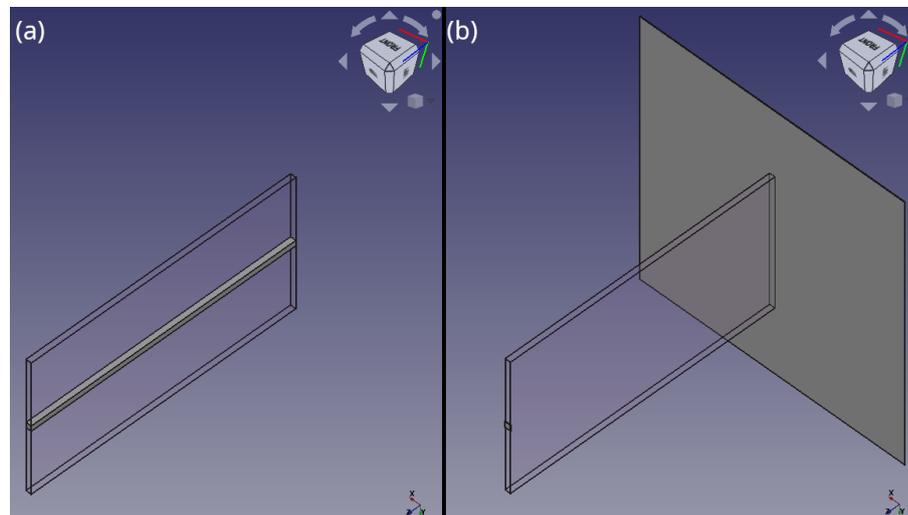


FIGURE 9.29. (a) CAD screenshot displaying the input object—bar and optimization domain. (b) CAD screenshot displaying the stressed area (left), supported area (right) and optimization domain.

The mesh from the structural optimization was directly used as an input for the feature recognition system. Refinement was run with the parameters displayed in Table 9.6. The refinement progress is shown in Figure 9.30.

TABLE 9.6. Parameters of feature recognition algorithm used in Case Study 1.

Parameter	Value
Initial cylinder radius	10 mm
Minimal Initial Element Size	40 mm
Population	5000
Node Move Standard Deviation	5.0 mm
Radius Change Standard Deviation	1.0 mm

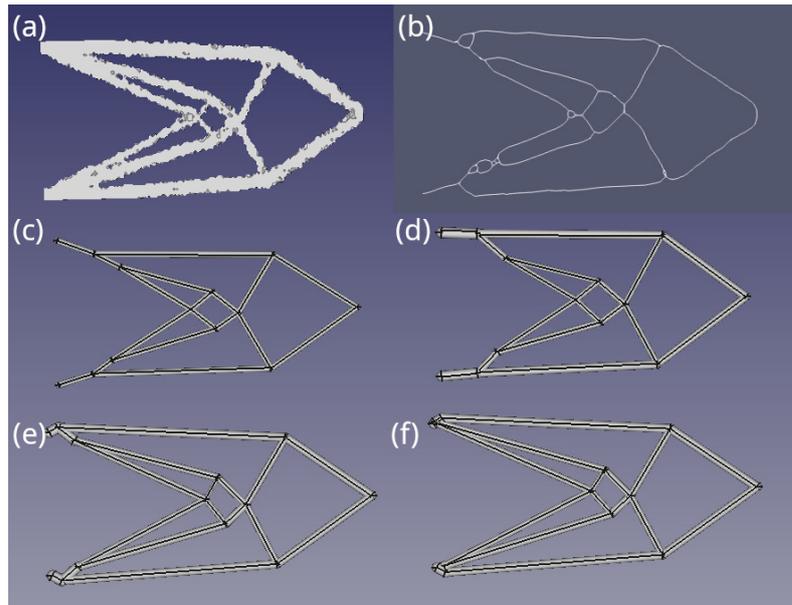


FIGURE 9.30. Initial mesh and refinement progress. Mesh in (a) is an STL file; (c–f) are CAD objects; (a) initial mesh; (b) output of skeletonization algorithm; (c) initial solution—input for evolutionary algorithm; (d) solution after 100 steps; (e) solution after 250 steps; (f) final solution after 564 steps.

The result of the feature recognition algorithms was imported into the CAD system and exported as an STL file, which was used in the solution verification (Figure 9.31). The numeric results are in Table 9.7. The boxplot of the surface von Mises stress is Figure 9.32. The reconstructed object is 0.3% lighter than the structural optimization result, so the algorithm produced an object of similar mass. The reconstructed object has a lower maximal surface von Mises stress and exhibits smaller deflection under load. The overall von Mises distribution is more concentrated (interquartile range  $19.97 \times 10^6$  Pa in reconstructed vs.  $52.67 \times 10^6$  Pa for reconstruction input), which suggests better distribution of load under stress in the reconstructed objects. This is because the compression/tension stresses dominate in this structure and the cylinder, which is the chosen structural element, carries those loads well. On the other hand, the von Mises stress is higher and a visual inspection of the FEM results shows excessive stress at the element connections (Figure 9.33). Those are the points where the bending moments concentrate. Disregarding the impossible load conditions, the implementation of the result would require manual strengthening of those points in CAD software to avoid structural failure.

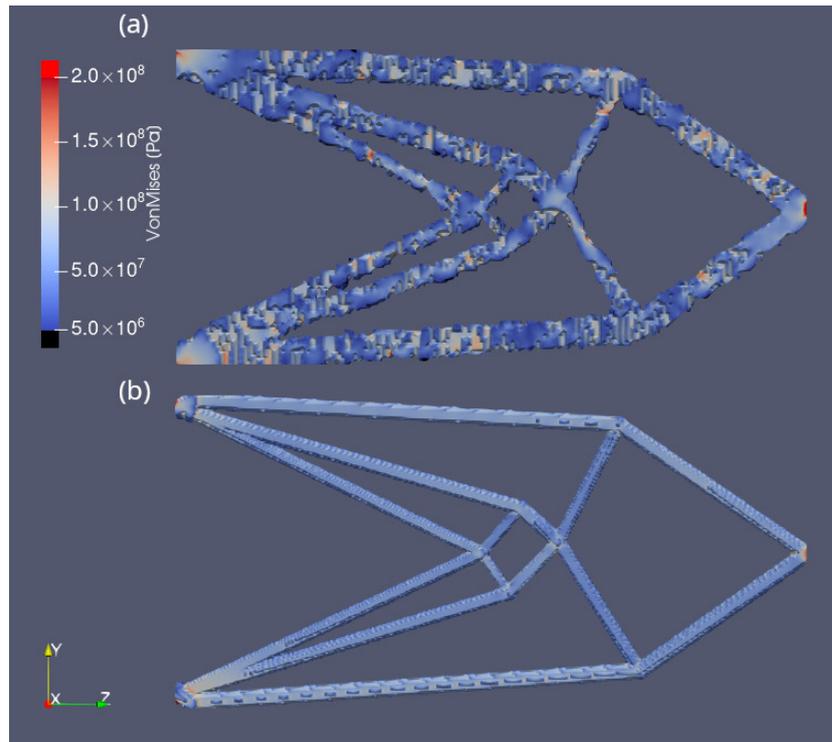


FIGURE 9.31. Comparison of von Mises stress for structural optimization output (a) and object reconstructed in CAD (b).

TABLE 9.7. Comparison of mass and stress values for input mesh and parametric model for Cast Study 1.

	<b>Input Mesh</b>	<b>Parametric Model</b>
Total Mass	18.41 kg	18.35 kg
Max von Mises stress on surface	$551 \times 10^6$ Pa	$436 \times 10^6$ Pa
Mean von Mises stress on surface	$55.7 \times 10^6$ Pa	$62.6 \times 10^6$ Pa
Max Deflection	2.008 mm	1.340 mm

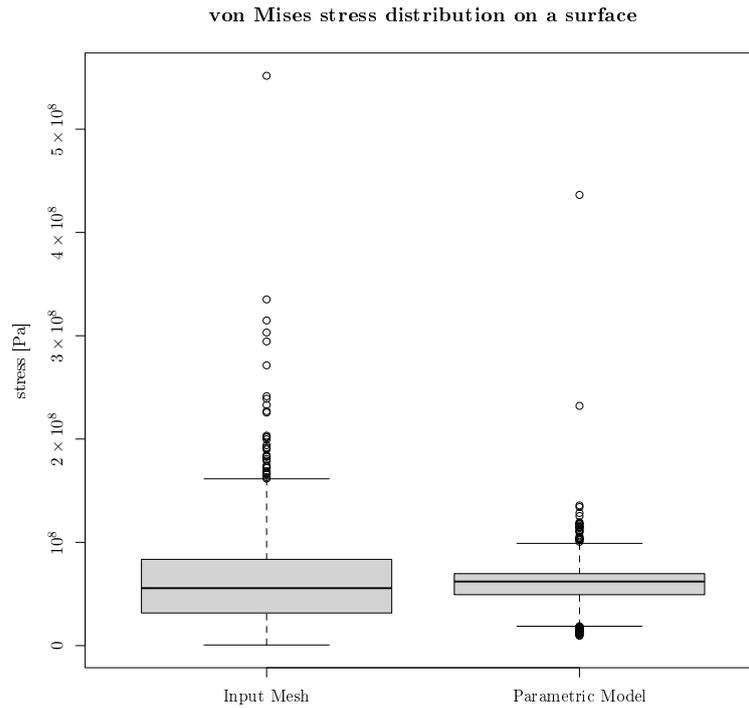


FIGURE 9.32. Boxplot showing von Mises stress distribution on a surface, calculated for both the feature recognition input mesh and the output parametric model. Compared to the input mesh, the distribution of the von Mises stress on a surface of the parametric model is more concentrated, but the mean value is higher.

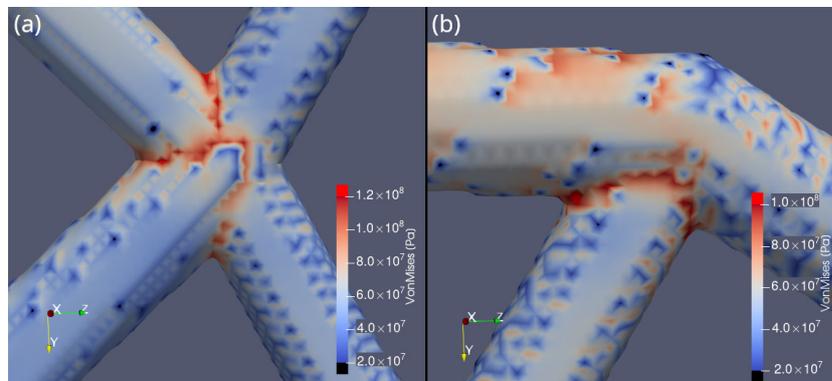


FIGURE 9.33. Two selected fragments of the object reconstructed in CAD exhibiting excessive von Mises stress at joints—bending moments under load are not sufficiently handled. (a) Central node of the mesh. (b) Node with tension element (bottom) and two compression elements (left and right).

### 9.3.3 Case Study 2—Box Corner

The second analyzed case study is a feature recognition run for the structural optimization of an aluminum box corner. The structural optimization has been performed as part of the mechanical design workflow. The manufacturing technique is 3D printing using the Powder Bed Fusion (PBF) technique. For manufacturing pre-processing, at the moment of this paper's creation, surface smoothing is considered. The following case study has been performed to evaluate the reconstruction of a parametric model as a pre-processing step for use in this particular case. The structural optimization using aluminum ( $\nu = 0.33$ ,

$E = 70 \times 10^9$  Pa,  $\rho = 2800$  kg/m<sup>3</sup>) was performed with multiple (three) load scenarios. The maximum allowed von Mises stress was set to  $70 \times 10^6$  Pa. The input geometries are shown in Figure 9.34. The boundary areas are color coded and used to describe the multiple load scenarios in Table 9.8. The optimization was run for 200 steps and the results of the optimization are shown in Figure 9.35. The prototype manufactured for the initial evaluation is in Figure 9.36.

TABLE 9.8. Forces and support assignment to boundary areas for load cases in the structural optimization task. Refer to Figure 9.34 for color-coded boundary areas' locations.

	Yellow Boundary	Blue Boundary	Red Boundary
<b>Load Case 1</b>	Force 200N, vector Y=-1	Force 200N, vector X=1	support
<b>Load Case 2</b>	Force 200N, vector Z=1	support	Force 200N, vector X=-1
<b>Load Case 3</b>	support	Force 200N, vector Z=1	Force 200N, vector Y=-1

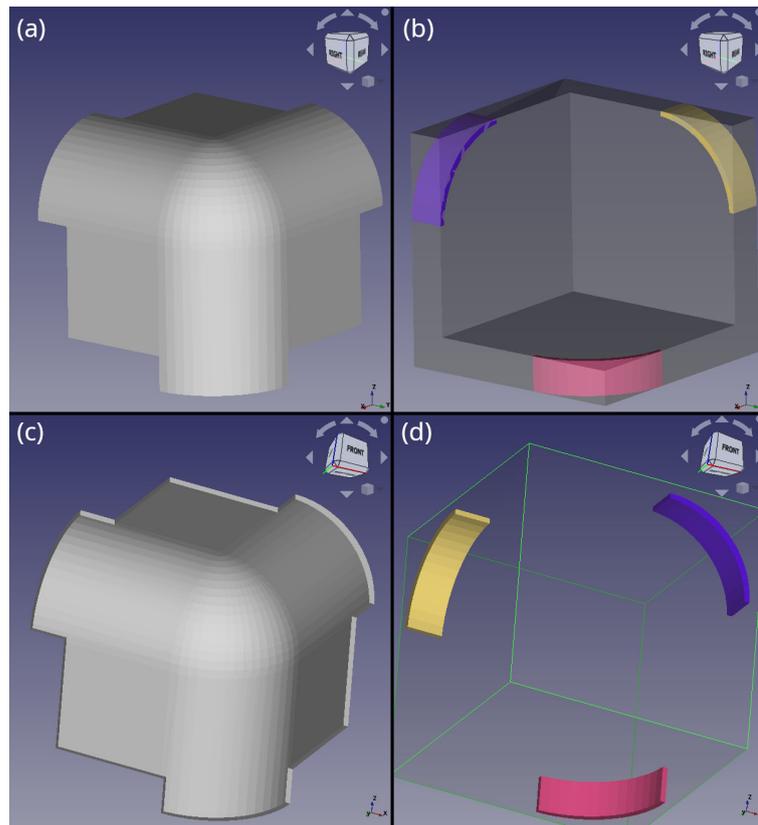


FIGURE 9.34. Definition of the structural optimization system input shapes in the CAD system. (a,c) Two different views of the element to be optimized. (b) Optimization exclusion area and three boundary areas. (d) Optimization domain and three boundary areas. Colors of boundary areas are used in load cases descriptions.

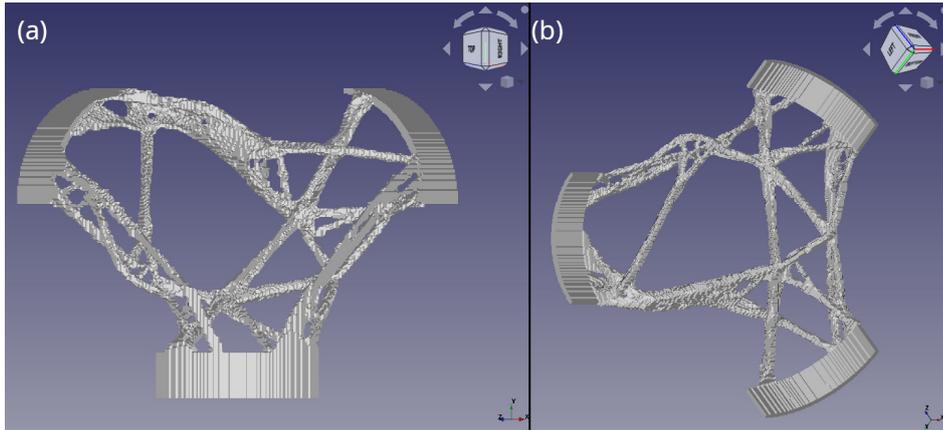


FIGURE 9.35. (a,b) Two different views of the topology optimization output—the input for the features recognition system.

TABLE 9.9. Parameters of feature recognition algorithm used in Case Study 2.

Parameter	Value
Initial Cylinder Radius	1.0 mm
Minimal Initial Element Size	3.0 mm
Population	5000
Node Move Standard Deviation	0.2 mm
Radius Change Standard Deviation	0.2 mm



FIGURE 9.36. Photograph of one of the prototypes of the optimized box corner. This particular element has been processed with surface smoothing [18] and manufactured using FDM technology. Aluminum box edges are attached to the optimized object's edges—areas excluded from optimization.

The mesh from the structural optimization was directly used as an input for the feature recognition system. Unlike in Case Study 1, this optimization problem contains areas excluded from optimization. Those areas were removed from the mesh in the pre-processing step. Refinement was run with the parameters displayed in Table 9.9. The output from the feature recognition process steps is shown in Figure 9.37. Structural analysis was performed after the parametric model was obtained, and areas excluded from optimization

were reattached to the reconstructed object in a CAD system. The altered mesh was exported from the CAD system in an STL format and used for the structural analysis. As the optimization problem contains three different load scenarios, the structural analysis was also performed three times, once for each load scenario. Table 9.10 contains a summary of the analysis. Figure 9.38 shows the surface stress distribution for each load scenario, which are also presented as boxplots (Figure 9.39).

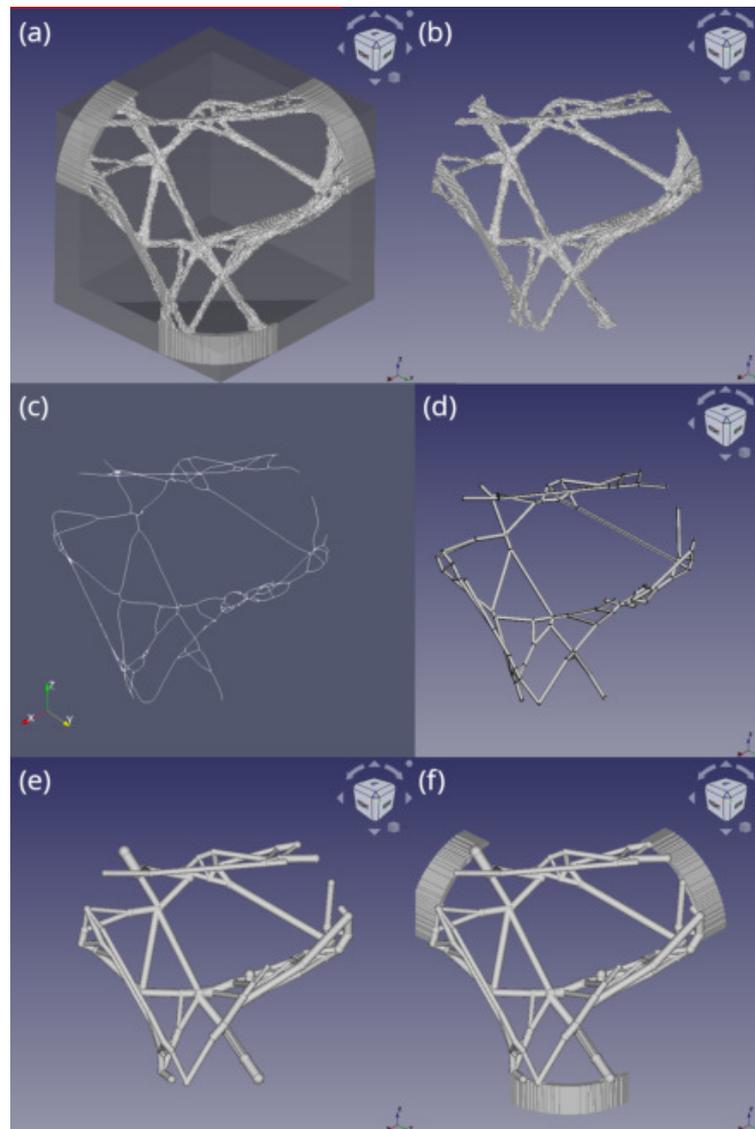


FIGURE 9.37. Steps of feature recognition algorithm. (a) Structural optimization output mesh with optimization exclusion area. (b) Structural optimization output mesh with parts inside optimization exclusion area removed. (c) Skeletonized mesh. (d) Initial solution—input of heuristic refinement as CAD features. (e) Output of heuristic refinement algorithm as CAD features. (f) CAD features with parts inside optimization exclusion area re-attached for solution validation.

TABLE 9.10. Comparison of mass and stress values for input mesh and parametric model for Case Study 2.

	<b>Input Mesh</b>	<b>Parametric Model</b>
Mass	64.1g	72.4g
<b>Load Case 1</b>		
Max Von Mises stress on surface	$141 \times 10^6 Pa$	$236 \times 10^6 Pa$
Mean Von Mises stress on surface	$23.1 \times 10^6 Pa$	$21.9 \times 10^6 Pa$
Max Deflection at Force 1	0.82mm	0.78mm
Max Deflection at Force 2	0.61mm	0.62mm
<b>Load Case 2</b>		
Max Von Mises stress on surface	$178 \times 10^6 Pa$	$248 \times 10^6 Pa$
Mean Von Mises stress on surface	$19.98 \times 10^6 Pa$	$17.6 \times 10^6 Pa$
Max Deflection at Force 1	0.73mm	0.61mm
Max Deflection at Force 2	0.68mm	1.0mm
<b>Load Case 3</b>		
Max Von Mises stress on surface	$263 \times 10^6 Pa$	$436 \times 10^6 Pa$
Mean Von Mises stress on surface	$28.3 \times 10^6 Pa$	$28.9 \times 10^6 Pa$
Max Deflection at Force 1	1.6mm	2.29mm
Max Deflection at Force 2	0.96mm	1.28mm

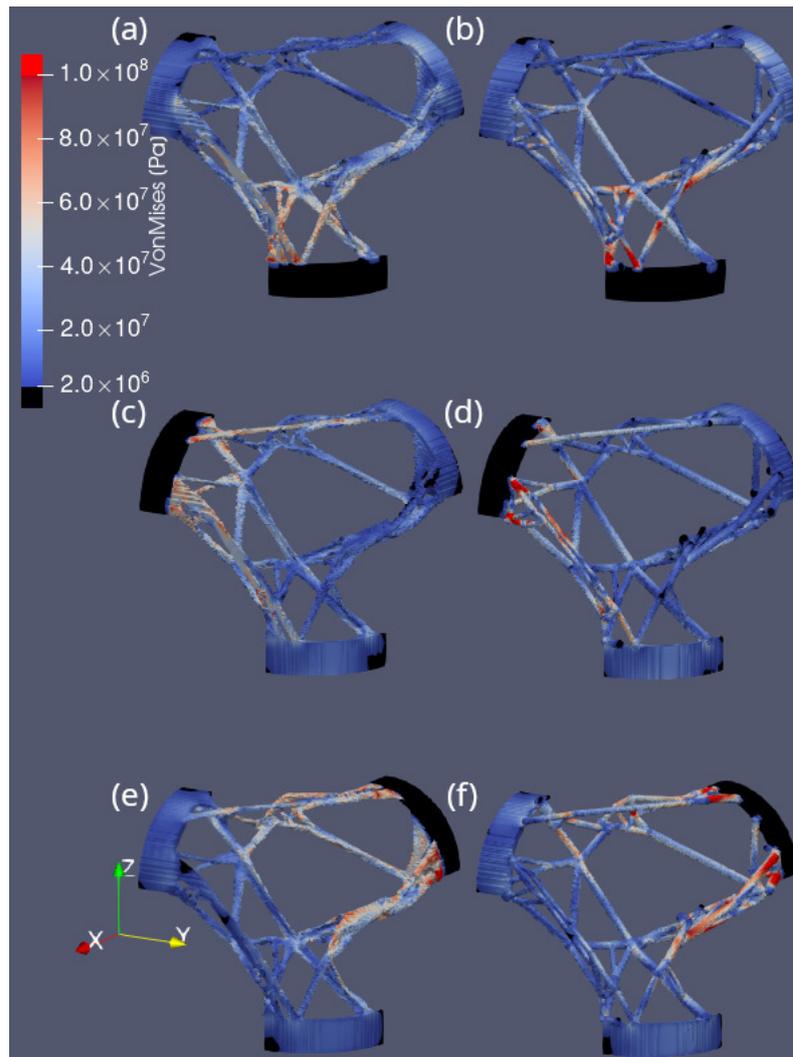


FIGURE 9.38. (a,c,e) von Mises stress on a surface calculated for the structural optimization output. (b,d,f) von Mises stress on a surface calculated for the mesh reconstructed in CAD; (a,b) Load Case 1. (c,d) Load Case 2. (e,f) Load Case 3.

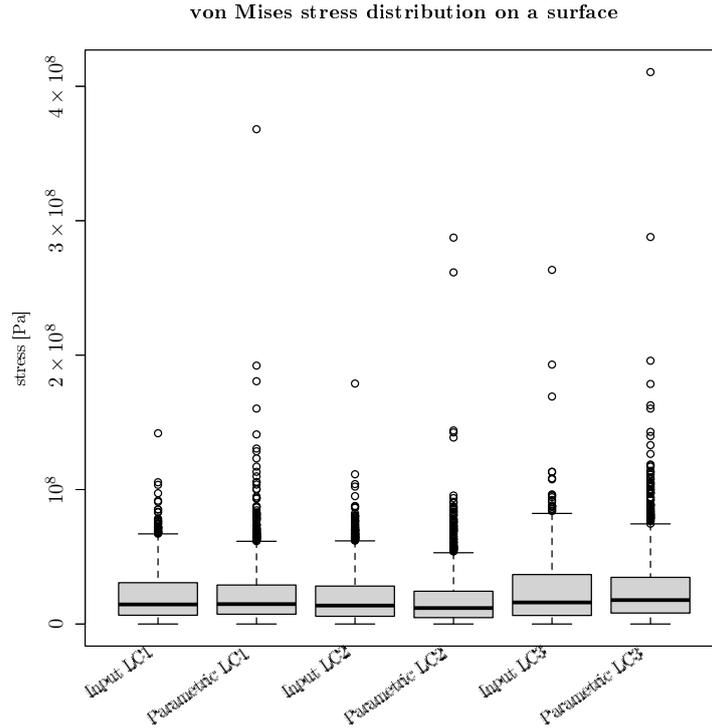


FIGURE 9.39. Boxplot showing von Mises stress distribution on a surface, calculated for both the feature recognition input mesh and the output parametric model, for each of three load cases. For each load case, the parametric model exhibits comparable mean stress values and the third quartile is lower, indicating better mechanical properties. On the other hand, the parametric model contains multiple outliers with high stress concentration. In practice, those areas determine the overall stiffness of the final design and are undesirable.

The reconstructed object exhibits both a higher overall mass and worse material properties. Although, for Load Case 1, the parametric model exhibits a lower mean stress and comparable deflection values. For Load Case 2 and Load Case 3, all analyzed measures exceed acceptable values. Details with areas exceeding the allowed surface von Mises stress are visible in Figure 9.40. Analysis shows that a cylindrical structural element, used to build the parametric model, is insufficient to represent the complex structure of the element optimized using the biomimetic approach. The surface stress exceeds the material's yield strengths in a few different distinct cases: the attachment points to the boundary areas are insufficient, the connection between elements are too weak and the cylinders exhibit poor behavior under bending moments (Figure 9.41). On the other hand all those problems can be solved manually by engineers in CAD software based on the provided structural analysis results.

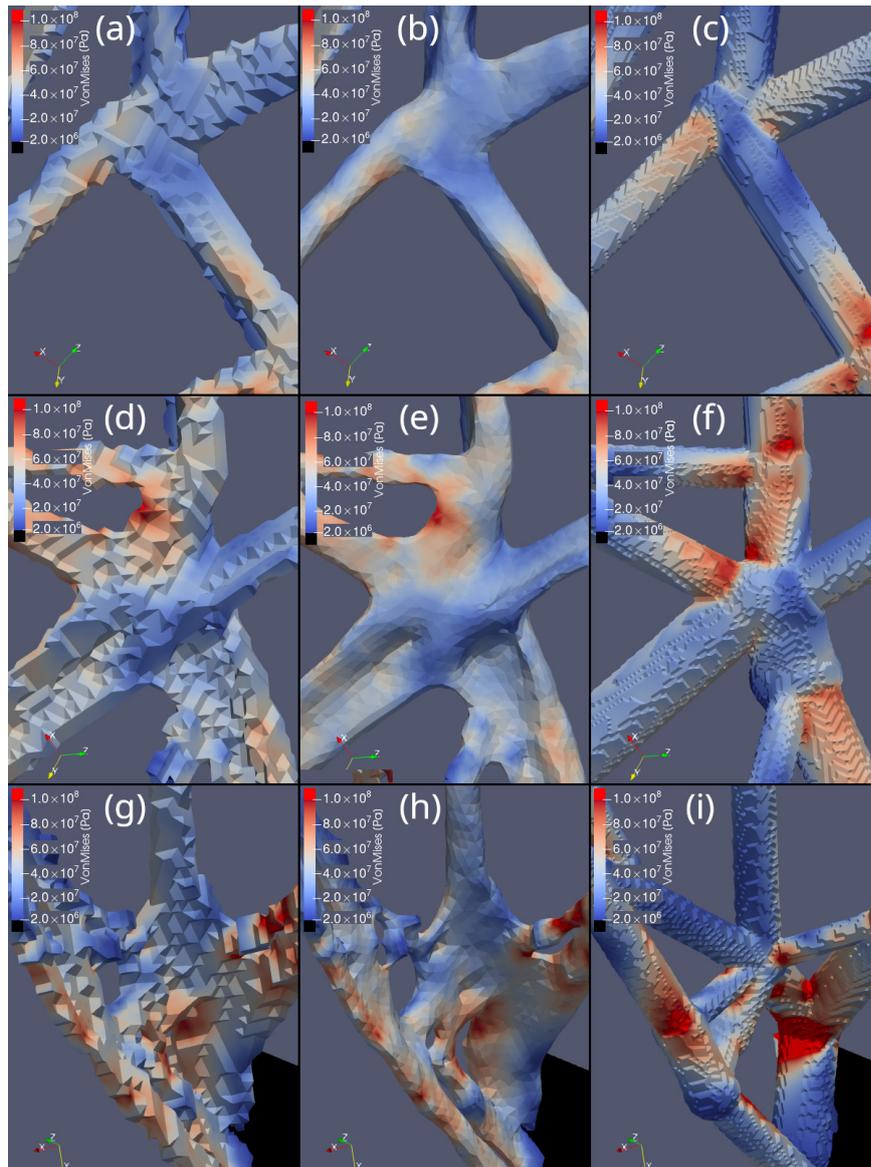


FIGURE 9.40. Comparison of structural analysis results between the structural optimization output with the reconstructed parametric model. (a,d,g) Structural optimization output. (b,e,h) Structural optimization output processed with surface smoothing for better visibility of the details. (c,f,i) Reconstructed parametric model. Three different details of the model are presented. Parts with the von Mises surface stress exceeded in the reconstructed model, compared to structural optimization output, resemble tapered cylinders. An approximation with cylinders is insufficient.

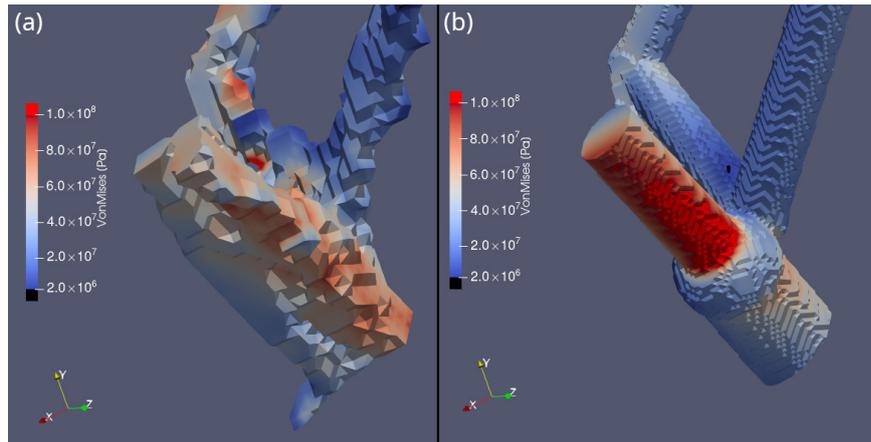


FIGURE 9.41. Mesh clipped to show stress concentration at Load Case 2 conditions at connection point between structural elements. (a) Structural optimization output. (b) Reconstructed parametric model. Flattened bar that carries bending moments, in structural optimization results it bears distinctive resemblance to the I-beam, which is designed for this purpose. In the parametric model, due to limitations, it was approximated with an element that is not suited for such loads.

## 9.4 Application of the proposed workflow to the industrial use case.

The workflow proposed in the Chapter8 was applied to the industrial use case. Case study was performed using structural optimization output of a box corner. The input of the optimization process was the same as described in the Section9.3. This particular result was selected because, as a part of the BioniAMoto project, optimization was created as a part of demonstration of the end-to-end manufacturing design process and the optimized mesh was manually prepared for manufacturing using LM-PBF technique. This gives the opportunity to execute a designed workflow and compare results with the mesh created manually. In the following section all steps of the proposed workflow are completed and results presented. The final mesh is then compared with mesh processed manually using mesh visualization software. For the input of the structural optimization process refer to Figure 9.34. The only difference in this task compared to the referred optimization task is a change of maximum allowed von Mises stress. In this optimization task this value was increased to  $75 \times 10^6$  Pa resulting in different topology. Optimization results with stress distribution calculated for one of three load scenarios is shown in a Figure 9.42. Mesh directly from the optimization was used as the input for the described workflow. The additional requirement, imposed by the manufacturing process, is requirement of minimal element radius of  $1.5mm$ . This limitation is not caused by manufacturing procedure directly (LM-PBF is able to produce finer details) but by use of support elements in the manufacturing process. Cutting off supports in the final product from small details may cause elements to break hence the structural optimization team was asked to thicken those elements during the mesh preparation process. All described steps were performed manually, using the FreeCAD software. Internally FreeCAD uses CGAL library [84–86] to perform mesh operations hence is possible to fully automate those steps. Meshing step can be automated using CGAL [87] or gmsh [66] but research on the exact strategy wasn't part of the scope of this work.

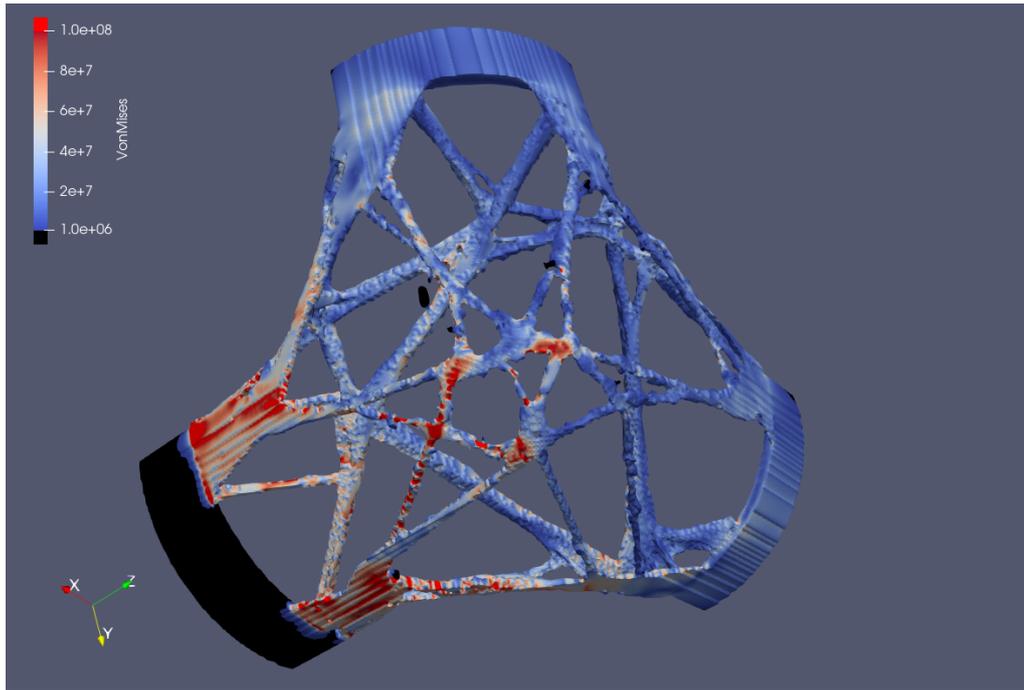


FIGURE 9.42. The result of the biomimetic structural optimization system – input of the processing workflow.

The first step of the proposed workflow, after the structural optimization is completed, is the extraction of the area of interest. This operation was performed using remeshing with the structural optimization system mesh generator, to avoid mesh issues described in the Chapter 7. In the same step secondary mesh was generated. Boolean mesh operation was used to calculate intersection of the original CAD input object with locked areas. Both meshes are shown in Figure 9.43. In this step the first unexpected issue was encountered. The initial shape shown in a panel (a) and excluded area in a panel (c) of Figure 9.43 share border which results in additional elements of zero volume produced by the intersection operation. Those elements were manually removed from elements shown on a panel (b). This issue is caused by incorrectly defined input data – to avoid numeric issues elements used in mesh boolean operations should always overlap slightly.

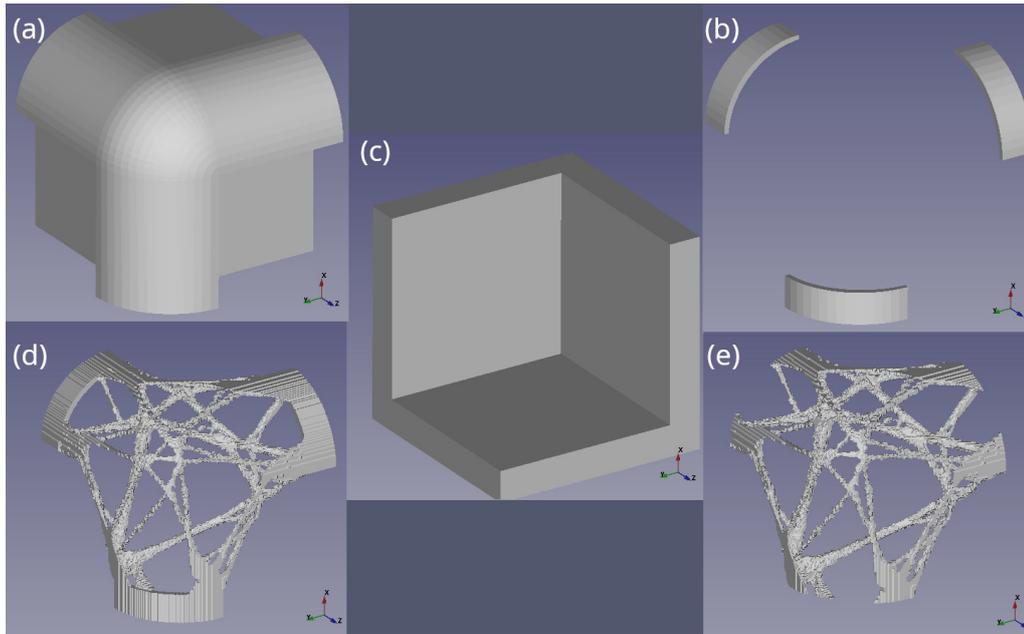


FIGURE 9.43. The initial step of mesh processing. Shape defining areas excluded from the optimization is used to remove those areas from optimized shape and to extract excluded shapes for later re-addition. (a) Input CAD model. (b) Mesh containing elements from excluded areas. (c) A shape defining area excluded from the optimization. (d) Optimization result. (e) Optimization result with removed areas excluded from optimization.

After mesh preparation workflow forks into classical processing with surface smoothing and parametric model extraction. Surface smoothing was performed by applying 500 iterations of the Laplacian Smoothing algorithm. Parametric model was prepared using an automated approach presented in the Chapter 7. Algorithm was run once, using default parameters. After the refinement step finished additional filtering was applied – all cylinders in the output mothe with a radius lower than  $1.5\text{mm}$  were modified to increase the radius to  $1.5\text{mm}$ . Additional operation, to prevent previously experienced numerical issues, was added. The radius of changed cylinders was increased by additional random value taken from uniform distribution between  $[0, 0.01]$ . During initial tests fusing cylinders of the same radius may lead to numeric issues and invalid output. Adding a small random number eliminates this unwanted phenomenon without visibly changing the solution. Smoothed mesh from the first path, the corrected parametric model from the second path and intermediate steps from the algorithm are in Figure 9.44. Manual Model Validation step, proposed as the last step of the parametric model path, was performed and a minor issue was spotted, shown on a panel (f) in Figure 9.44. Issue was caused by the proposed refinement algorithm itself. The goal is to fill the input mesh with available elements and the algorithm doesn't change topology. On the other hand, the parametric model allows easy correction of such issues as shown on a panel (g) in Figure 9.44. Correction of this particular issue took below 3 minutes of work in a CAD software. After the parametric model was verified and corrected it was converted back into finite element mesh using the Netgen algorithm.

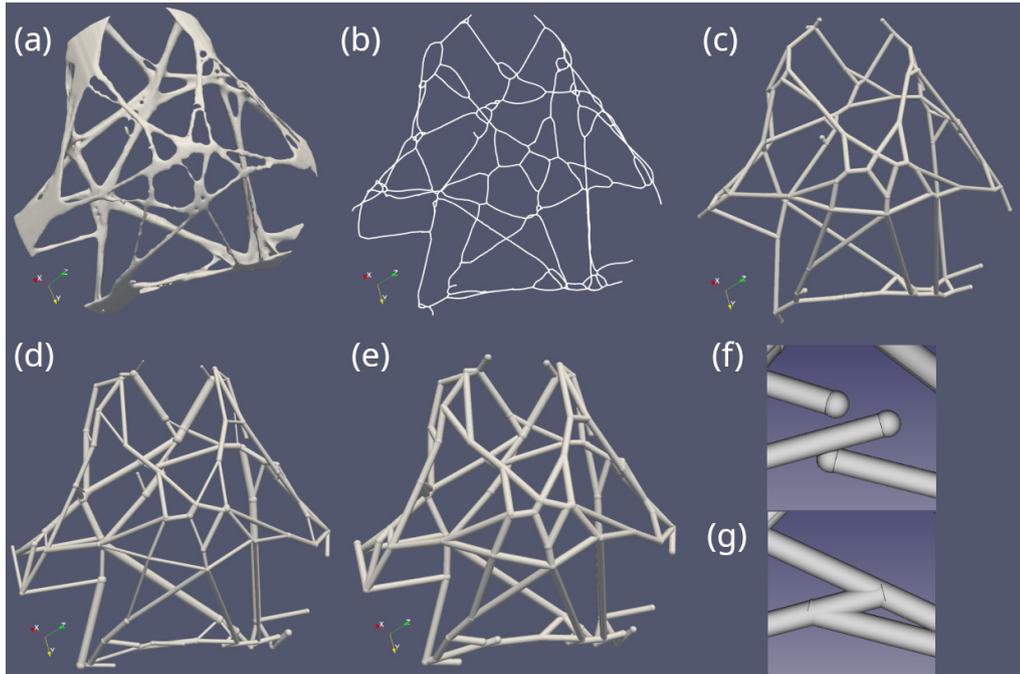


FIGURE 9.44. The initial step of mesh processing. Shape defining areas excluded from the optimization are used to remove those areas from optimized shape and to extract excluded shapes for later re-addition. (a) Mesh smoothed with 500 iterations of the Laplacian Smoothing algorithm. (b) Skeleton of the input mesh. (c) The initial parametric solution. (d) The refined parametric solution. (e) The parametric solution with applied limitations of the manufacturing process. (f) Issue in the parametric model found during the Manual Model Validation step. (g) Corrected issue in the parametric model.

The final step is a merge of all 3 meshes – smoothed, parametric and with areas excluded from optimization. During the mesh fusing step minor issue with the smoothed mesh was identified. Smoothing causes volume shrink, so connection point between smoothed mesh and excluded areas was incomplete in the fused mesh. This in practice would lead to a structural failure. To prevent this all 3 elements in excluded areas were manually extended by 1mm towards the optimized mesh. Complete mesh is shown in Figure 9.45. The whole process described required less than 15 minutes of work to finish, not including the initial job preparation time and waiting time for the optimization and for the feature recognition to finish. The fully manual processing procedure took 2 hours to finish.

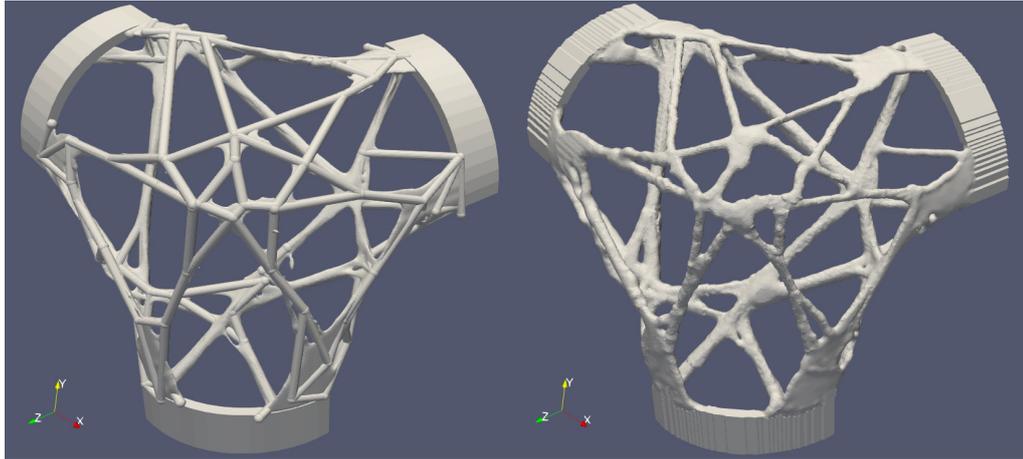


FIGURE 9.45. The structural optimization result processed to include limitation of the manufacturing technique – minimal required radius of elements is  $1.5\text{mm}$ . (a) Mesh generated using the proposed processing workflow. (b) Mesh processed manually by a technician.

Compared to the manually processed mesh, the mesh obtained using hybrid workflow exhibits a significantly smoother surface. Synergistic properties of the proposed approach are visible in internal nodes of the mesh as shown in Figure 9.46. Uniform thickness of structural elements provides superior load-bearing characteristics under tension/compression compared to non uniform items, where the whole element has strength of its smallest cross-section. Judging from the mesh shape and prevalence of straight bars this kind of stress dominates in structure. To verify this hypothesis structural analysis was performed for both meshes using the first load scenario, results are shown in Figure 9.47. Structural analysis supports the observation – smooth, cylindrical elements generated by the skeletonization-based feature recognition approach exhibit more uniform stress distribution thus carry loads better than manually processed elements of the mesh.

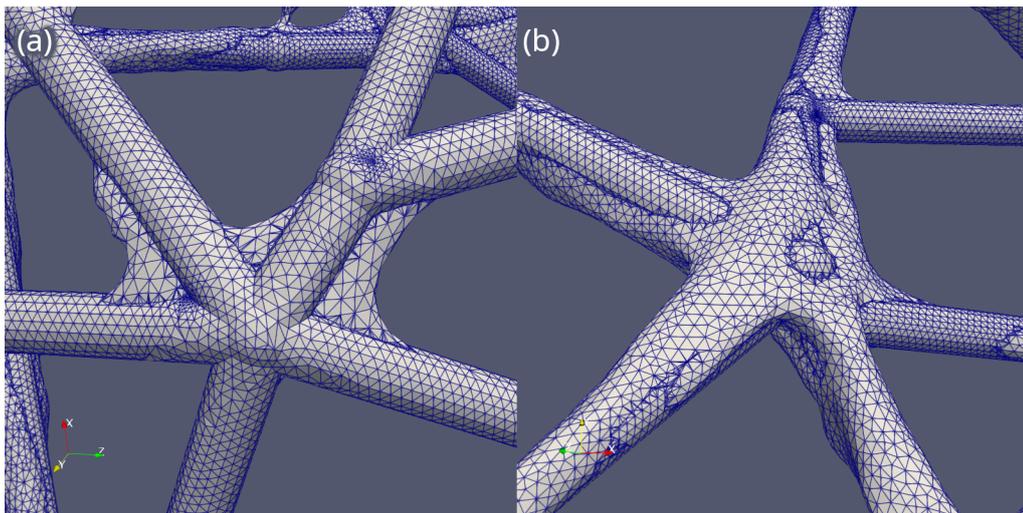


FIGURE 9.46. (a, b) Details of the mesh obtained using the proposed workflow. Bionic structure was supplemented by a parametric model.

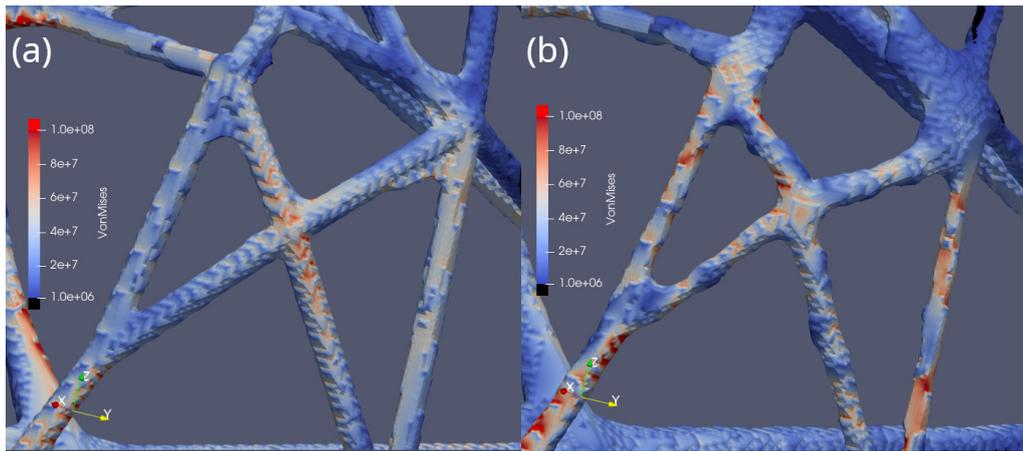


FIGURE 9.47. Detail of structural optimization output showing von Mises stress on a surface for the first load case from the optimization task. (a) Mesh processes with hybrid workflow. (b) Mesh manually processed. Smooth structures obtained from the parametric model provide better load distribution compared to the manually processed mesh.

# Chapter 10

## Discussion

This chapter reflects on the aim of the thesis, discussing achieved results and the proposed advancements in the technologies and what shortcomings have been identified. The primary aim was to research tools for building parametric geometric models for CAD systems based on topological optimization results. The aim was achieved in distinct steps. First mesh generator of the biomimetic structural optimization system was evaluated and shortcomings were addressed to enable further research. Then two different approaches for the truss recognition problem were researched. Finally, an industrialization approach using hybrid workflow was proposed.

### 10.1 Mesh generator tool

The primary aim of this research was to evaluate current state and propose improvements to allow use of the mesh generator both to generate an input for the parametric model recognition system and a part of it. Proposed changes were successfully implemented and validated.

Mesh generated using a researched system now looks smoother and requires less processing before manufacturing, when surface smoothing algorithms are used. Meshes still have rough surfaces for geometries containing elements angled at low angles, due to scale reduction step. The use of the Marching Cubes algorithm [88, 89] for mesh generation would remove this issue. The reason why a custom mesh generator is implemented is to ensure uniform structural element size across the whole mesh to improve stability of FEM structural analysis – the primary purpose of mesh building. Enhancing the structural optimization system with a secondary, more detailed, mesh generator would allow producing better quality meshes for further processing while retaining uniform element size of elements for FEM structural analysis. On the other hand using a different mesh for the output than the one optimized requires additional validation steps. In this context the benefit can be not worth the overall complication of the optimization process. More research is required to address this issue.

Meshes generated by the structural optimization system are now free of defects. It was surprising to learn that all FEM software used during the research (Calculix, Elmer FEM, FrontISTR) are able to produce valid results using meshes with minor topology issues.

On the other hand those issues prevent any operations with mesh processing software, FreeCAD and underlying CGAL in particular. Although it is possible to fix those meshes using the dedicated software (MeshMixer, FreeCAD) it was shown complex topologies can't be automatically fixed. Moreover researched algorithms are based on element removal which decreases volume of the mesh. Moreover it was observed that mesh issues often appear in thin structural elements, removal of faulty edges may split such elements in half changing mesh topology. Hence it was justified to undertake the effort of addressing those issues in the mesh generator itself. The ability to produce meshes of higher quality enables new research directions for the structural optimization system itself.

## 10.2 Hough Transform

Cylinder recognition algorithm using Hough Transform based recognition was proposed and implemented. Tests were conducted using both synthetic and real data. Suitability of this approach for processing 2D results is well proven. For 3D data cylinders recognition with a two step approach was evaluated. The first stage is cylinder axis estimation, the second is radius and position estimation. The algorithm was implemented as proof-of-concept, with all shortcomings identified and listed. Validation procedure of the output parametric model was proposed. The implementation was proven to work properly on simple test cases, then the algorithm was tested on results of the biomimetic structural optimization algorithm. All results were compared with source data and usability of the algorithm was evaluated.

The main reason for selecting Hough transform for the first research was the formal elegance of this approach. Input data, after transformation into a feature space, is changed into a representation easy to interpret by both humans and the algorithm. Two-stage algorithm allows future extensions as each part can operate separately with separate parts for axis detection and separate for shape estimation. Transformation into the feature space is fast and, despite the algorithm uses multidimensional accumulators, memory requirements are low considering current computer capabilities, especially compared to the amount of memory required to perform structural optimization.

The algorithm works well for synthetic data. As the second stage of the algorithm is designed to detect circles, other shapes cause detection errors as shown in both the second case study and in tests on structural optimization outputs. For the proof-of-concept work achieved quality of detection is considered sufficient. Extending detection to other shapes, such ellipses, is possible using Hough transform [90], but in practical application the second step doesn't need to use this approach for detection. Once data is projected onto a plane along the element axis any algorithm can be used to recognize the shape. Visual inspection of the projected data suggests estimation of the axis position can be performed using points grouping algorithms followed by the radius estimation based on a simple fitting algorithm such as minimizing mean square error. The algorithm provides no means to estimate quality of the detection and stop criterion is not defined. More research is required to provide such capabilities. The identified issue with detection of cylinders with axis along the Z axis has been mitigated by picking test cases without such data.

Tests performed on the real data, on the other hand, demonstrated poor performance of the algorithm. Simple truss meshes are recognized properly, but more complex topologies are either recognized partially or not at all. The algorithm is very sensitive to the data quality and struggles with flat areas of mesh which interfere with the axis estimation step resulting in false-positive detection errors. Correction of such issues is hard, as demonstrated on the 4th mesh in the case study 3. This property makes use of this algorithm for the feature recognition in a standalone more problematic. Real data contains different features and the algorithm has a tendency to “stick” to those problematic areas instead of detecting features it can correctly detect.

To sum up, provided implementation is correct but detection quality for the real data is insufficient. Moreover there is no easy way to extend the algorithm to either provide capabilities of detection of other geometries or to increase the quality of current recognition. Further research on this algorithm, as an end-to-end solution for the feature recognition, is not recommended. On the other hand the researched approach based on the multi-stage processing including parts of the algorithm can be valuable in the further research. Each estimation step operates independently. Axis estimated with different approaches, including researched skeletonization, can be processed with a radius estimation algorithm to improve the quality of the initial solution.

### 10.3 Skeletonization-based approach

Skeletonization-based approach was extensively researched. The thesis presents algorithm description and provides test results performed both on well-recognized literature test cases and for the real data, obtained in the BioniAMoto project. Test results include structural analysis using FEM software and the analysis of suggested quality measures.

The proposed skeletonization-based approach with topology extraction using a skeletonization algorithm, followed by filtering and heuristic refinement works really well for both, as demonstrated with the output from the BioniAMoto project. Truss structures produced by the biomimetic structural optimization system can be easily identified by a skeletonization algorithm.

Skeleton filtering step, the first step after skeletonization, is based on explicitly provided parameters, minimal edge length in particular. In all performed tests value was assumed *a priori*. For the box corner estimation value of  $3mm$  was used, which corresponds to about 2.5% of the bounding box size. There is no recommendation for the right value of this parameter, which is critical to the overall performance of the process. Too low value may result in a needlessly complex initial solution which, paired with an evolutionary algorithm used for the heuristic refinement, leads to noisy final solution when small elements are used to fill available space in a chaotic manner instead of creating a neat truss. Too high minimal element size leads to loss of details.

The initial solution is far from optimal and requires a heuristic refinement step. Initial solution can be improved by performing the radius estimation for each element in the initial solution. This can be achieved using the Radius Estimation step from the researched Hough Transform approach.

Heuristic refinement step relies on the Fitness Function to score results and find the best next candidate. The proposed formula was extensively researched and performs well. The biggest identified issue is operation on the volumetric data completely disregarding mesh topology. This makes possible very fast computation but leads to results where elements are not connected or are lightly touching each other. In practice such structures should be connected together. Second issue is the problem arising from the underlying philosophy of the proposed Fitness Function. It scores how well the solution fits the mesh, not how well the mesh is represented by the solution. If some part of the input data is not initially included in the solution, i.e. because the skeletonization filtering step removed too many edges, those parts will never be included in the solution. Current implementation doesn't change the mesh topology - the graph of the initial solution is kept intact through the refinement process. For the proof of concept work this is sufficient, but the production-ready solution should be more robust, with the ability to remove not needed edges and trace curved structures by splitting edges.

Provided algorithm struggles with a space filling task in parts of the mesh which are not truss. In the provided solution for the BioniAMoto box corner this is apparent near support areas where truss morphs into flat attachment points. The algorithm tries to fill those volumes with available elements creating chaotic mesh with bad stress-carrying capabilities. This weakness was well-understood during the design and implementation phase and was effectively mitigated by the proposed hybrid approach. Nevertheless it is recommended to address this shortcoming in further research by introducing more variety of structural elements. Change of the cylinder to the extruded ellipse as the basic structural elements appears to be the good direction of the next research step. Compared to a circle, an ellipse has 2 more degrees of freedom so it is able to provide more variety of forms without complicating the solution. Alternatively structural elements can be changed to the tapered cylinders by adding one more degree of freedom to the solution – the taper angle. Proposed solution, especially the robust Fitness Function, allows easy integration of additional geometries into the algorithm.

The described algorithm in current form, despite found shortcomings, proved useful in the BioniAMoto project where it was already used to prepare mesh for the additive manufacturing process. Further research and development is expected to allow desired degree of automation to enable use of the approach in industrial application outside of research projects.

## 10.4 Industrialization with the hybrid approach

Workflow enabling application of the research results in current form was proposed. Workflow is based on fusing an obtained parametric model back into the finite element mesh to create synergistic effect of both approaches. The extracted parametric model enables easy adjustment of the data in a CAD system to address additional requirements of the selected manufacturing process. Smoothed mesh, used for the fusion, helps to retain the bionic nature of the result mesh and balances details loss caused by the selected truss-based approach. Additive nature of changes required in the post processing step make it

possible to use simple fusion operation of meshes, which would not be possible if some parts of mesh are removed in a CAD model. In addition to two mentioned meshes, the third fused object are parts of the mesh excluded from optimization. Proposed workflow extracts them from the original CAD input of the structural optimization system hence those items are not distorted by the data discretization algorithm used by the optimization.

The proposed workflow was applied to the part developed during the BionIAMoto project. It was demonstrated how the process both speeds up overall processing and reduces effort required. Resulting mesh is considered to be of higher quality than manually prepared, allows repeatability of results and makes possible to perform processing using the standard CAD software, without mesh sculpting knowledge. Moreover it was proved with the structural analysis tools that smooth, straight surfaces of structural elements from the parametric model exhibit better load carrying properties than manually prepared mesh or smoothed mesh. The proposed workflow provides a unique view on the optimization result post processing technique and the research should be continued.

# Chapter 11

## Conclusions and future work plan

This chapter summarizes the main features of the proposed framework, discusses general features and specific research contributions, comments on how well overall research objectives were satisfied, and gives recommendations for further research.

### 11.1 Conclusions

This dissertation describes the framework for building parametric geometric models for CAD systems based on topological optimization results. The input for the system is a biomimetic structural optimization system. Mesh generator used by the system was evaluated and adjusted to provide high quality mesh enabling feature recognition. Two approaches were discussed - Hough Transformation application for cylinders recognition and skeletonization-based feature recognition. Both approaches were designed to transform the input mesh into a set of pre-defined simple geometries which can be transferred to the CAD system. Based on the research on the feature recognition algorithm the complete processing workflow based on the hybrid approach was proposed. All discussed approaches were implemented and tested using both synthetic and real data. The main novel contributions of this research are summarized as:

- The biomimetic structural optimization system was enriched with a better mesh generator. It enables further research in the area of structural optimization using this system enabling higher quality results using more detailed meshes. Use of this system is not limited to the area of mechanical design, but also can be used in the medical research areas as it allows trabecular bone modeling and analysis.
- Explored capabilities of the Hough Transform in the mechanical design and structural optimization areas are the first attempt, to the author's knowledge, of such applications.
- Proposed skeletonization-based algorithm provides a unique view of the feature recognition decomposing 3D mesh into a set of simple geometries. It defines the direction of next research, supported with successful application in an industrial use case.

- Hybrid approach to structural optimization results post processing is a novel approach for the mesh preparation for a manufacturing. Proposed framework bridges the gap between the structural optimization and manufacturing step allowing semi-automated application of manufacturing limitation to the optimization results. Research went beyond theoretical analysis of the dissertation aim and resulted in the industrial application of results, as a mesh that is ready to manufacture.

To conclude, the presented research provides significant advancement in the mechanical design workflow. The hybrid approach workflow, paired with a skeletonization-based feature recognition system, enables easy processing of the structural optimization results to prepare the model for the manufacturing step. The original research achievement of the work are:

- Skeletonization-based feature recognition system using evolutionary algorithm for data refinement,
- Hybrid workflow for optimization results preparation by fusing the adjusted parametric model with the original mesh.

## 11.2 Future Work

The described process, although already used in industrial application in a research project, is a framework enabling further development. The main areas of improvement are as follows:

- Truss-based geometry was chosen because, compared to more complex geometries, it has fewer degrees of freedom to optimize, so the heuristic refinement process converges fast. The chosen structural elements are insufficient for structures with bending moments under load and there is a need for support of more complex geometries to properly capture the complexity of the input meshes. The proposed heuristic refinement step and the Fitness Function in particular work with solutions built out of any parameterized geometries. This makes it possible to apply the algorithm to any geometries produced by structural optimization algorithms, without the current limitation to trusses.
- The refinement process in the skeletonization-based approach is heuristic; hence, the final solution may not meet the required mechanical properties. There is a need for research into the viability of adding an additional final refinement step where feedback from structural analysis is used to adjust the model. Heuristic refinement in this proof of concept work utilizes a random search approach. This process can be highly optimized. For example, adjusting positions of solution graph nodes is possible with the use of gradient descent methods. The initial topology, obtained from the processed mesh skeleton, is not changed during refinement. It is possible to enrich the heuristic refinement step with operations by changing the solution graph topology.

- Hough transform approach was considered to be too sensitive to data noise to work correctly in the presented use case. Parts of the research, the radius estimation algorithm in particular, can be used in the skeletonization-based approach to improve the quality of the initial solution. The approach with projecting the data to the plane normal to the detected feature axis allows an easy way to use 2D computer vision algorithms to extract information. On the other hand sensitivity of this approach to axis estimation errors was shown. In the skeletonization-based approach those estimation errors are completely ignored as refinement step fixes all initial problems. More research is required to determine if it is possible to fuse both approaches.
- The proposed workflow is able to consider only the most simple limitation – the minimal element size – which is easy to enforce. Another kind of processing is adding material to remove a need for support structures. Automated mesh processing to perform this task based on the manufacturing method would greatly improve capabilities of the researched solution.

# List of Figures

3.1	Data Discretization and Volumetric Mesh Building algorithms applied for sample image. Image is divided into squares, in this particular example size of 3 was selected. For each square, four points are selected and used for volumetric mesh building. Last image shows a volumetric mesh created from two identical images. . . . .	11
3.2	Showcase of the multiple load case and mesh evolution algorithm. On the left element with random holes subjected to shear stress (top) and compression (bottom). In the middle both cases are aggregated using formula 3.2. On the right the same element after the evolution algorithm has been applied and then both load cases are calculated and aggregated together. New material 'grows' in high and is removed from low strain energy density areas. Material with strain energy density values inside the lazy zone is left intact. . . . .	11
3.3	Sample input for the structural optimization system, as seen in a CAD system. Gray – the main element. Green – three distinct boundary areas. While outline – optimization domain bounding box. . . . .	12
3.4	Sample visualization of multiple load scenario definition. Defined boundary areas can have assigned different boundary conditions in each load case scenario allowing flexible optimization task definition. In the shown example <i>boundary_left.stl</i> is supported in the <i>Multiload Case1</i> and stressed area in <i>Multiload Case2</i> . . . . .	13
3.5	Sample output mesh of the biomimetic structural optimization system, obtained for the input data from Figure 3.3. (a) Optimized mesh. (b) Optimized mesh with smoothing applied, for visualization purposes only. . . . .	13
5.1	Data Discretization and Volumetric Mesh Building algorithms applied for a sample image. The image is divided into squares, in this particular example size of 3 was selected. For each square, four points are selected and used for the volumetric mesh building. The last image shows volumetric mesh created from two identical images. . . . .	17
5.2	Subset of data used to test the mesh generator. Data comes from the micro-CT of a rat femur bone and was selected because of complexity of the resulting mesh. . . . .	18
5.3	Mesh of the rat femur bone created from the micro-CT scan. Cubical structural elements, each containing 6 tetras are clearly visible. . . . .	18

5.4 The result of the mesh analyzed with the FreeCAD system. Multiple non-manifolds were found. . . . . 20

5.5 Mesh details with multiple non-manifolds highlighted in red. . . . . 20

5.6 Showcase of the mesh repair feature in FreeCAD. (a) Mesh details with non-manifolds. (b) Mesh after using the repair option. Invalid edges were removed leaving open holes in the mesh. (c) Mesh after using the "fill holes" algorithm in FreeCAD, as recommended in the manual. Algorithm added just a single face, and wasn't able to correctly fix the mesh. . . . . 21

5.7 Smoothing of the mesh surface using the Laplacian Smoothing algorithm. (a) The input mesh. (b) Detail of the input mesh (c) After 200 iterations of Laplacian Smoothing. Rough surface is still noticeable. (d) After 500 iterations of Laplacian Smoothing. Rough surface is noticeable but smoothness can be considered sufficient. (e) Detail from (b) after 500 iterations of Laplacian Smoothing. Significant reduction of volume is visible. . . . . 21

5.8 (a) In a correctly built mesh each faces connectivity subgraph created from faces sharing a given edge contains exactly one cycle. (b) Detail of a mesh with the single point non-manifold. When the faces connectivity subgraph is constructed for faces sharing given point, there exist two separate closed paths allowing detection. . . . . 22

6.1 Example of the Hough transform for line detection. Left: Input data. Right: Hough space accumulator obtained for the data on the left. By Daf-de - Own work, CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=1121165>. . . . . 23

6.2 Two faces of a cylinder surface with corresponding circles in Hough space. Each circle corresponds to all possible axes of cylinders that include a given face. The color of the face corresponds to the color of the circle generated of its normal. Intersection of two circles corresponds to the actual axis of the input cylinder. Data comes from the actual algorithm output. Faces were obtained from triangulation using gmsh software with the 'Frontal Delaunay' generator [64,65]. . . . . 25

6.3 Hough space accumulator for the mesh generated with the MeshAdapt algorithm [66] with an overlaid input cylinder, shrunk down to match the size of the Gaussian sphere. Point with maximal accumulator value corresponds to the cylinder axis. . . . . 26

6.4 Cylinder surface points projected to the plane normal to the cylinder axis. . . 27

6.5 Two slices of Hough space accumulator used for the position and radius estimation. Accumulator is filled using a random subsample of 5% of the input data. (a) The slice corresponding to the radius of 4.5mm. (b) The slice corresponding to the radius of 5.0mm. When radius is correctly estimated, all circles in the accumulator converge in a single point, which corresponds to the right radius and position. . . . . 28

7.1 Activity diagram of the feature recognition algorithm. Preprocessing and skeletonization steps lead to an initial solution, which is refined using a heuristic approach. . . . . 32

7.2 Example of feature recognition algorithm input. **(a,b)** Input of the structural optimization algorithm, as seen in the CAD system. **(a)** Optimized object and optimization domain boundary. **(b)** Optimized object and three boundary condition areas, which in this particular case are also areas excluded from optimization. Bottom: structural optimization output. **(c)** Raw surface mesh. **(d)** Surface mesh smoothed, only for visualization purposes. . . . . 32

7.3 Impact of mesh density on skeletonization algorithm. **(a)** Mesh preprocessed with CGAL algorithms to perform removal of areas excluded from optimization. **(b)** Skeleton generated for **(a)** mesh. One of the structural elements completely collapsed. **(c)** Mesh with removed areas excluded from optimization using mesh generator from structural optimization system. Mesh retained vertex density. **(d)** Skeleton generated for **(c)** mesh. . . . . 34

7.4 **(a,c)** Example output meshes from topology optimization system. **(b,d)** Skeletons generated for corresponding meshes. . . . . 35

7.5 Justification for skeleton filtering: single detail of the structural optimization output and corresponding skeleton for 3 consecutive optimization steps. **(a,c,e)** Details of structural optimization output with skeletons drawn over. **(b,d,f)** Details on the left with smoothing applied, for visualization purposes only, with skeletons drawn over. Detail represents an unstable region of the optimization domain—the porous structure appears and disappears because a mesh density is too low to converge to the stable state. . . . . 35

7.6 **(a)** Optimized cantilever beam. Mesh was created out of a 2D image by extruding it. Skeleton generated for the shown mesh is drawn over. **(b)** Initial solution generated from processed skeleton. Two skeleton edges were below the minimum length threshold and collapsed. Bottom-right node of the solution graph was generated by splitting an edge of the skeleton graph. . . . . 36

7.7 Detail of the input mesh surrounded by guide points (pink). Fitness Function yields positive score for aligning solution to the mesh surface and gives penalty for including guide points. . . . . 37

7.8 **(a)** Extracted features exported to CAD as cylinders. Partial connections between elements are visible. **(b)** The same features exported as cylinders with additional spheres at the ends. All elements are fully connected. . . . . 39

8.1 Proposed hybrid workflow integrating structural optimization process with additive manufacturing using both mesh smoothing and feature recognition approach. Highlighted manual steps of the workflow. . . . . 44

9.1 The result of analysis with FreeCAD Mesh Evaluator of the mesh generated for micro-CT of rat bone. Analysis indicated no mesh issues. . . . . 46

9.2 Showcase of the non-manifold repair algorithm on a mesh detail. **(a)** The original mesh, the non-manifold edge highlighted in red. **(b)** The corrected mesh. Additional volume in the area of the non-manifold edge was added. . . . 47

9.3 Detail of a mesh generated from rat bone micro-CT showing the quality of input data mapping. **(a)** Mesh created with the original mesh generator. **(b)** Mesh created with the new mesh generator. Addition of new structural elements enables better mapping of the input data into the output mesh. . . . . 47

9.4 Detail of a mesh generated from rat bone micro-CT showing the surface quality of a mesh. **(a)** Mesh created with the original mesh generator. **(b)** Mesh created with the new mesh generator. Addition of new structural elements allows better surface mapping. . . . . 48

9.5 Mesh detail, same as in Figure 9.4 after 100 iterations of the Laplacian Smoothing algorithm. **(a)** Mesh created with the original mesh generator. **(b)** Mesh created with the new mesh generator. Smoother surface of the input data results helps to achieve better results after smoothing. . . . . 48

9.6 Parametric model of the element used for the first test. Sketch on the left was extruded by rotation around the vertical axis to obtain the element on the right. 49

9.7 **(a)** Mesh generated for the parametric model of the test cylinder using the default export settings in FreeCAD software. The side surface consists of big, high aspect ratio elements while the ends contain very detailed, small elements. **(b)** Detail of the cylinder end. Faces of varied size, with high aspect ratio, are visible. . . . . 50

9.8 Meshes for the parametric cylinder generated using **(a)** Mefisto Mesher and **(b)** MeshAdapt algorithms. Both meshes were generated for element size = 0.4mm. 50

9.9 Meshes for the parametric cylinder generated using the mesh generator from the structural optimization system. The parametric model exported as an STL file was used as an input. **(a)** Direct output. **(b)** Output with Laplacian Smoothing algorithm applied. . . . . 51

9.10 Hough space accumulators obtained for the first step of the algorithm - the direction estimation. **(a)** Mefisto Mesher. **(b)** MeshAdapt. **(c)** Structural Optimizer Mesh Generator. **(d)** Structural Optimizer Mesh Generator with Laplacian smoothing. Detection based on Structural Optimizer Mesh Generator output is impossible but smoothing mesh fixes this problem. . . . . 52

9.11 **(a)** Magnification of Hough space accumulator area corresponding to the cylinder axis obtained for the mesh generated with the Mefisto Mesher algorithm. Circular region indicates that the mesh generated with this algorithm contains some issue. **(b)** Magnification of the surface of the cylinder mesh generated with the Mefisto Mesher algorithm. Red line is the surface of the parametric cylinder. Visual inspection indicates surface faces are slightly angled which explains the obtained pattern in the accumulator. . . . . 53

9.12 Parametric object used in Case Study 2. The shape was designed to test the feature recognition algorithm on a more complex data than a simple cylinder before testing on a real data. Big and small cylinders are not collinear, there is 3° of a difference between axes. . . . . 54

9.13 A detail of the input mesh. **(a)** Directly from the structural optimizer mesh generator. **(b, c, d)** Post-processed using the Laplacian smoothing algorithm with 100, 200 and 500 iterations respectively. As a surface smoothness increases fine details are lost. . . . . 54

9.14 Results of the feature detection for the input mesh processed with the Laplacian smoothing using **(a)** 100 iterations, **(b)** 200 iterations, **(c)** 500 iterations. Input parametric model in light gray, reconstructed parametric model in dark gray. The biggest feature in **(a)** consists of two elements. The horizontal feature with an elliptical cross-section in both **(b)** and **(c)** was detected as a part of a bigger cylinder instead. . . . . 55

9.15 Progress of feature detection algorithm for the mesh after 500 iterations of the Laplacian Smoothing. The hough space accumulator for the initial state **(a)** and after the first iteration **(c)**. Initial mesh as a points cloud **(b)** and mesh as a points cloud after the first iteration **(d)**. Removal of the feature detected during the first step uncovered another feature, previously hidden under votes generated by the recognised feature. . . . . 56

9.16 Progress of the feature detection algorithm for the mesh after 500 iterations of the Laplacian Smoothing. The hough space accumulator for the state before the third iteration **(a)** and after the third iteration **(c)**. Mesh before the third iteration as a points cloud **(b)** and mesh as a points cloud after the third iteration **(d)**. . . . . 57

9.17 Test data used for feature recognition in Test Case 3. All meshes were obtained using the structural optimization algorithm for the cantilever beam under different input conditions. All meshes were pre-processed using the Laplacian smoothing algorithm. **(a, b, c)**: Test mesh 1. **(d, e, f)**: Test mesh 2. **(g, h, i)**: Test mesh 3. **(j, k, l)**: Test mesh 4. **(a, d, g, j)**: View of test meshes along the Z axis. **(b, e, h, k)**: View of test meshes along the X axis. **(c, f, i, l)**: View of test meshes along the Y axis. . . . . 58

9.18 **(a, b)**: Parametric model obtained for the Mesh 1. **(c, d)**: Input mesh (white) overlapping reconstructed parametric model (black). Although the input mesh looks symmetric to the naked eye, the feature recognition process uncovered differences in the thickness between elements. . . . . 59

9.19 Feature recognition results for the Mesh 2. **(a, b)** Input mesh. **(c)**, **(d)** Parametric model. **(e, f)** Input mesh (white) and parametric model (black) showing feature recognition details. The first recognised feature was the small cylinder on the top-right part of **(d)** - both direction and radius were recognised correctly. Other three features have right axis direction but radius detection algorithm failed. Slightly curved surfaces of the input mesh were detected as parts of big circles. . . . . 60

9.20 Mesh surface points projected on the plane normal to the detected cylinder axis (white dots) and the cylinder estimated using this data (gray) for the third iteration of the algorithm running for the Mesh 2. . . . . 61

9.21 Feature recognition results for the Mesh 3. **(a, b)** Input mesh. **(c, d)** Parametric model. **(e, f)** Input mesh (white) and parametric model (black) showing feature recognition details. The same problem as with the Mesh 2 appeared, despite some features being recognised correctly. Note how the flat area on the left side of the mesh **(a)** was recognised as a cylinder with incorrect axis. . . . . 62

9.22 Feature recognition results for the Mesh 3 limited to correctly detected features. **(a, b)** Parametric model. **(c, d)** Input mesh (white) and parametric model limited to correctly recognised features (black). . . . . 63

9.23 **(a, b)**: Parametric model obtained for the Mesh 4. **(c, d)**: Input mesh (white) overlapping reconstructed parametric model (black). Algorithm failed completely. Flat sides of the mesh interfered with the radius estimation algorithm leading to wrong results. Algorithm failed to extract any usable knowledge from the input data. . . . . 64

9.24 **(a)** The mesh used for the feature recognition, obtained from the structural optimization result mesh by applying 500 iterations of the Laplacian Smoothing algorithm. **(b)** The structural optimization result mesh after applying 2000 iterations of the Laplacian Smoothing algorithm. Increased smoothing increases curvature of flat areas, but distorts fine details and greatly decreases overall mesh volume. . . . . 64

9.25 An attempt to increase the algorithm accuracy by manually removing flat sides from the input data. **(a)** Input mesh. **(b)** Input mesh (white) overlaid with the parametric model (black). After removal of mesh sides there are not enough data points to perform the feature recognition. . . . . 64

9.26 Bremicker’s cantilever beam [75] processed by the feature recognition algorithm. **(a)** Structural optimization result. **(b)** Skeletonized mesh. **(c)** Initial parametric solution. **(d)** Final refined parametric solution. . . . . 65

9.27 Gedig’s cantilever beam [6] processed by the feature recognition algorithm. **(a)** Structural optimization result. **(b)** Skeletonized mesh. **(c)** Initial parametric solution. **(d)** Final refined parametric solution. . . . . 66

9.28 **(a,c)** Mesh of Bremicker’s cantilever beam [75] (light) overlaid with the reconstructed parametric model (dark). **(b,d)** Mesh of Gedig’s cantilever beam [6] (light) overlaid with the reconstructed parametric model (dark). Both shown cases visually demonstrate the quality of feature recognition results. In both cases, the reconstructed parametric solution matches the shape of the input. . . 66

9.29 **(a)** CAD screenshot displaying the input object—bar and optimization domain. **(b)** CAD screenshot displaying the stressed area (left), supported area (right) and optimization domain. . . . . 67

- 9.30 Initial mesh and refinement progress. Mesh in **(a)** is an STL file; **(c–f)** are CAD objects; **(a)** initial mesh; **(b)** output of skeletonization algorithm; **(c)** initial solution—input for evolutionary algorithm; **(d)** solution after 100 steps; **(e)** solution after 250 steps; **(f)** final solution after 564 steps. . . . . 68
- 9.31 Comparison of von Mises stress for structural optimization output **(a)** and object reconstructed in CAD **(b)**. . . . . 69
- 9.32 Boxplot showing von Mises stress distribution on a surface, calculated for both the feature recognition input mesh and the output parametric model. Compared to the input mesh, the distribution of the von Mises stress on a surface of the parametric model is more concentrated, but the mean value is higher. . . . . 70
- 9.33 Two selected fragments of the object reconstructed in CAD exhibiting excessive von Mises stress at joints—bending moments under load are not sufficiently handled. **(a)** Central node of the mesh. **(b)** Node with tension element (bottom) and two compression elements (left and right). . . . . 70
- 9.34 Definition of the structural optimization system input shapes in the CAD system. **(a,c)** Two different views of the element to be optimized. **(b)** Optimization exclusion area and three boundary areas. **(d)** Optimization domain and three boundary areas. Colors of boundary areas are used in load cases descriptions. . . . . 71
- 9.35 **(a,b)** Two different views of the topology optimization output—the input for the features recognition system. . . . . 72
- 9.36 Photograph of one of the prototypes of the optimized box corner. This particular element has been processed with surface smoothing [18] and manufactured using FDM technology. Aluminum box edges are attached to the optimized object's edges—areas excluded from optimization. . . . . 72
- 9.37 Steps of feature recognition algorithm. **(a)** Structural optimization output mesh with optimization exclusion area. **(b)** Structural optimization output mesh with parts inside optimization exclusion area removed. **(c)** Skeletonized mesh. **(d)** Initial solution—input of heuristic refinement as CAD features. **(e)** Output of heuristic refinement algorithm as CAD features. **(f)** CAD features with parts inside optimization exclusion area re-attached for solution validation. . . . . 73
- 9.38 **(a,c,e)** von Mises stress on a surface calculated for the structural optimization output. **(b,d,f)** von Mises stress on a surface calculated for the mesh reconstructed in CAD; **(a,b)** Load Case 1. **(c,d)** Load Case 2. **(e,f)** Load Case 3. . . . . 75
- 9.39 Boxplot showing von Mises stress distribution on a surface, calculated for both the feature recognition input mesh and the output parametric model, for each of three load cases. For each load case, the parametric model exhibits comparable mean stress values and the third quartile is lower, indicating better mechanical properties. On the other hand, the parametric model contains multiple outliers with high stress concentration. In practice, those areas determine the overall stiffness of the final design and are undesirable. . . . . 76

9.40 Comparison of structural analysis results between the structural optimization output with the reconstructed parametric model. **(a,d,g)** Structural optimization output. **(b,e,h)** Structural optimization output processed with surface smoothing for better visibility of the details. **(c,f,i)** Reconstructed parametric model. Three different details of the model are presented. Parts with the von Mises surface stress exceeded in the reconstructed model, compared to structural optimization output, resemble tapered cylinders. An approximation with cylinders is insufficient. . . . . 77

9.41 Mesh clipped to show stress concentration at Load Case 2 conditions at connection point between structural elements. **(a)** Structural optimization output. **(b)** Reconstructed parametric model. Flattened bar that carries bending moments, in structural optimization results it bears distinctive resemblance to the I-beam, which is designed for this purpose. In the parametric model, due to limitations, it was approximated with an element that is not suited for such loads. . . . . 78

9.42 The result of the biomimetic structural optimization system – input of the processing workflow. . . . . 80

9.43 The initial step of mesh processing. Shape defining areas excluded from the optimization is used to remove those areas from optimized shape and to extract excluded shapes for later re-addition. **(a)** Input CAD model. **(b)** Mesh containing elements from excluded areas. **(c)** A shape defining area excluded from the optimization. **(d)** Optimization result. **(e)** Optimization result with removed areas excluded from optimization. . . . . 81

9.44 The initial step of mesh processing. Shape defining areas excluded from the optimization are used to remove those areas from optimized shape and to extract excluded shapes for later re-addition. **(a)** Mesh smoothed with 500 iterations of the Laplacian Smoothing algorithm. **(b)** Skeleton of the input mesh. **(c)** The initial parametric solution. **(d)** The refined parametric solution. **(e)** The parametric solution with applied limitations of the manufacturing process. **(f)** Issue in the parametric model found during the Manual Model Validation step. **(g)** Corrected issue in the parametric model. . . . . 82

9.45 The structural optimization result processed to include limitation of the manufacturing technique – minimal required radius of elements is 1.5mm. **(a)** Mesh generated using the proposed processing workflow. **(b)** Mesh processed manually by a technician. . . . . 83

9.46 **(a, b)** Details of the mesh obtained using the proposed workflow. Bionic structure was supplemented by a parametric model. . . . . 83

9.47 Detail of structural optimization output showing von Mises stress on a surface for the first load case from the optimization task. **(a)** Mesh processes with hybrid workflow. **(b)** Mesh manually processed. Smooth structures obtained from the parametric model provide better load distribution compared to the manually processed mesh. . . . . 84

# List of Tables

9.1	Comparison of the source cylinder parameters with the results of the algorithm for three different mesh generators. . . . .	51
9.2	Proposed quality measures calculated for three input meshes. . . . .	52
9.3	Accuracy measures calculated for the Case Study 2 results for the mesh processed with 100 iterations of the Laplacian Smoothing algorithm . . . . .	55
9.4	Accuracy measures calculated for the Case Study 2 results for the mesh processed with 200 iterations of the Laplacian Smoothing algorithm. . . . .	55
9.5	Accuracy measures calculated for the Case Study 2 results for the mesh processed with 500 iterations of the Laplacian Smoothing algorithm. As the input mesh smoothing increases so does the axis detection accuracy, but volume shrinkage is noticeable as radiuses are underestimated. Element 2, with elliptical cross-section, is incorrectly estimated and error increases as more smoothing is applied. . . . .	56
9.6	Parameters of feature recognition algorithm used in Case Study 1. . . . .	67
9.7	Comparison of mass and stress values for input mesh and parametric model for Cast Study 1. . . . .	69
9.8	Forces and support assignment to boundary areas for load cases in the structural optimization task. Refer to Figure 9.34 for color-coded boundary areas' locations. . . . .	71
9.9	Parameters of feature recognition algorithm used in Case Study 2. . . . .	72
9.10	Comparison of mass and stress values for input mesh and parametric model for Case Study 2. . . . .	74

# Acronyms

API	Application Programming Interface
BMU	Basic Multicellular Unit
CAD	Computer-Aided Design
CNC	Computer Numerical Control
FEM	Finite Element Method
FDM	Fused Deposition Modeling
LC	Load Case
LM-PBF	Laser Metal Powder Bed Fusion
LP-PBF	Laser Polymer Powder Bed Fusion
PBF	Powder Bed Fusion
SIMP	Solid Isotropic Material with Penalization
SED	Strain Energy Density
SLS	Selective Laser Sintering
STL	Standard Triangle Language (file format)
2D	two-dimensional
3D	three-dimensional

# Bibliography

- [1] W. Wang, D. Munro, C. C. Wang, F. van Keulen, and J. Wu, “Space-time topology optimization for additive manufacturing: Concurrent optimization of structural layout and fabrication sequence,” *Structural and Multidisciplinary Optimization*, vol. 61, pp. 1–18, 2020.
- [2] Z. Jihong, Z. Han, W. Chuang, Z. Lu, Y. Shangqin, and W. Zhang, “A review of topology optimization for additive manufacturing: Status and challenges,” *Chinese Journal of Aeronautics*, vol. 34, no. 1, pp. 91–110, 2021.
- [3] Y. Saadlaoui, J.-L. Milan, J.-M. Rossi, and P. Chabrand, “Topology optimization and additive manufacturing: Comparison of conception methods using industrial codes,” *Journal of Manufacturing Systems*, vol. 43, pp. 178–186, 2017.
- [4] N. A. Sbrugnera Sotomayor, F. Caiazzo, and V. Alfieri, “Enhancing design for additive manufacturing workflow: optimization, design and simulation tools,” *Applied Sciences*, vol. 11, no. 14, p. 6628, 2021.
- [5] E. Dalpadulo, F. Pini, and F. Leali, “Integrated cad platform approach for design for additive manufacturing of high performance automotive components,” *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 14, no. 3, pp. 899–909, 2020.
- [6] M. Gedig, *A framework for form-based conceptual design in structural engineering*. PhD thesis, University of British Columbia, 2010.
- [7] A. Mandhyan, G. Srivastava, and S. Krishnamoorthi, “A novel method for prediction of truss geometry from topology optimization,” *Engineering with Computers*, vol. 33, no. 1, pp. 95–106, 2017.
- [8] T. Shannon, *A feature-mapping optimisation framework for CAD based design*. PhD thesis, Queen’s University Belfast, 2023.
- [9] J.-F. Gamache, A. Vadean, É. Noirot-Nérin, D. Beaini, and S. Achiche, “Image-based truss recognition for density-based topology optimization approach,” *Structural and Multidisciplinary Optimization*, vol. 58, pp. 2697–2709, 2018.
- [10] D. Brackett, I. Ashcroft, and R. Hague, “Topology optimization for additive manufacturing,” in *2011 International Solid Freeform Fabrication Symposium*, University of Texas at Austin, 2011.

- [11] M. Sehmi, J. Christensen, C. Bastien, and S. Kanarachos, "Review of topology optimisation refinement processes for sheet metal manufacturing in the automotive industry," *Structural and Multidisciplinary Optimization*, vol. 58, pp. 305–330, 2018.
- [12] M. Marinov, M. Amagliani, T. Barback, J. Flower, S. Barley, S. Furuta, P. Charrot, I. Henley, N. Santhanam, G. T. Finnigan, *et al.*, "Generative design conversion to editable and watertight boundary representation," *Computer-Aided Design*, vol. 115, pp. 194–205, 2019.
- [13] G. Taubin, "A signal processing approach to fair surface design," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 351–358, 1995.
- [14] J. Vollmer, R. Mencl, and H. Mueller, "Improved laplacian smoothing of noisy surface meshes," in *Computer graphics forum*, vol. 18, pp. 131–138, Wiley Online Library, 1999.
- [15] M. Wei, J. Huang, X. Xie, L. Liu, J. Wang, and J. Qin, "Mesh denoising guided by patch normal co-filtering via kernel low-rank recovery," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 10, pp. 2910–2926, 2018.
- [16] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 317–324, 1999.
- [17] K.-W. Lee and W.-P. Wang, "Feature-preserving mesh denoising via bilateral normal filtering," in *Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05)*, pp. 6–pp, IEEE, 2005.
- [18] O. Ibhádode, Z. Zhang, J. Sixt, K. M. Nsiempba, J. Orakwe, A. Martinez-Marchese, O. Ero, S. I. Shahabad, A. Bonakdar, and E. Toyserkani, "Topology optimization for metal additive manufacturing: current trends, challenges, and future outlook," *Virtual and Physical Prototyping*, vol. 18, no. 1, p. e2181192, 2023.
- [19] S. Larsen and C. G. Jensen, "Converting topology optimization results into parametric cad models," *Computer-Aided Design and Applications*, vol. 6, no. 3, pp. 407–418, 2009.
- [20] J. Liu and Y. S. Ma, "3d level-set topology optimization: a machining feature-based approach," *Structural and Multidisciplinary Optimization*, vol. 52, pp. 563–582, 2015.
- [21] C. Yeo, B. C. Kim, S. Cheon, J. Lee, and D. Mun, "Machining feature recognition based on deep neural networks to support tight integration with 3d cad systems," *Scientific reports*, vol. 11, no. 1, p. 22147, 2021.
- [22] J. Lee, H. Lee, and D. Mun, "3d convolutional neural network for machining feature recognition with gradient-based visual explanations from 3d cad models," *Scientific Reports*, vol. 12, no. 1, p. 14864, 2022.

- [23] T. Stangl, S. Wartzack, *et al.*, “Feature based interpretation and reconstruction of structural topology optimization results,” in *DS 80-6 Proceedings of the 20th International Conference on Engineering Design (ICED 15) Vol 6: Design Methods and Tools-Part 2 Milan, Italy, 27-30.07. 15*, pp. 235–244, 2015.
- [24] A. Amroune, J.-C. Cuillière, and V. François, “Automated lofting-based reconstruction of cad models from 3d topology optimization results,” *Computer-Aided Design*, vol. 145, p. 103183, 2022.
- [25] A. Parfitt, C. Mathews, A. Villanueva, M. Kleerekoper, B. Frame, D. Rao, *et al.*, “Relationships between surface, volume, and thickness of iliac trabecular bone in aging and in osteoporosis. implications for the microanatomic and cellular mechanisms of bone loss,” *The Journal of clinical investigation*, vol. 72, no. 4, pp. 1396–1409, 1983.
- [26] P. Ehrlich and L. E. Lanyon, “Mechanical strain and bone cell function: a review,” *Osteoporosis international*, vol. 13, no. 9, p. 688, 2002.
- [27] Z. Miller, M. B. Fuchs, and M. Arcan, “Trabecular bone adaptation with an orthotropic material model,” *Journal of biomechanics*, vol. 35, no. 2, pp. 247–256, 2002.
- [28] R. Müller, “Long-term prediction of three-dimensional bone architecture in simulations of pre-, peri-and post-menopausal microstructural bone remodeling,” *Osteoporosis International*, vol. 16, pp. S25–S35, 2005.
- [29] R. Müller, M. Kampschulte, T. E. Khassawna, G. Schlewitz, B. Hürter, W. Böcker, M. Bobeth, A. C. Langheinrich, C. Heiss, A. Deutsch, *et al.*, “Change of mechanical vertebrae properties due to progressive osteoporosis: combined biomechanical and finite-element analysis within a rat model,” *Medical & biological engineering & computing*, vol. 52, pp. 405–414, 2014.
- [30] I. Giorgio, U. Andreaus, D. Scerrato, and F. Dell’Isola, “A visco-poroelastic model of functional adaptation in bones reconstructed with bio-resorbable materials,” *Biomechanics and modeling in mechanobiology*, vol. 15, pp. 1325–1343, 2016.
- [31] I. Giorgio, M. Spagnuolo, U. Andreaus, D. Scerrato, and A. M. Bersani, “In-depth gaze at the astonishing mechanical behavior of bone: A review for designing bio-inspired hierarchical metamaterials,” *Mathematics and Mechanics of Solids*, vol. 26, no. 7, pp. 1074–1103, 2021.
- [32] J. Wolff, “Das gesetz der transformation der knochen,” *DMW-Deutsche Medizinische Wochenschrift*, vol. 19, no. 47, pp. 1222–1224, 1893.
- [33] R. Huiskes, H. Weinans, H. Grootenboer, M. Dalstra, B. Fudala, and T. Slooff, “Adaptive bone-remodeling theory applied to prosthetic-design analysis,” *Journal of biomechanics*, vol. 20, no. 11-12, pp. 1135–1150, 1987.
- [34] R. Huiskes, R. Ruimerman, G. H. Van Lenthe, and J. D. Janssen, “Effects of mechanical forces on maintenance and adaptation of form in trabecular bone,” *Nature*, vol. 405, no. 6787, pp. 704–706, 2000.

- [35] R. Huiskes, “If bone is the answer, then what is the question?,” *Journal of anatomy*, vol. 197, no. 2, pp. 145–156, 2000.
- [36] D. R. Carter, “Mechanical loading histories and cortical bone remodeling,” *Calcified tissue international*, vol. 36, no. Suppl 1, pp. S19–S24, 1984.
- [37] D. George, R. Allena, and Y. Remond, “Integrating molecular and cellular kinetics into a coupled continuum mechanobiological stimulus for bone reconstruction,” *Continuum Mechanics and Thermodynamics*, vol. 31, pp. 725–740, 2019.
- [38] I. Giorgio, F. Dell’Isola, U. Andreaus, F. Alzahrani, T. Hayat, and T. Lekszycki, “On mechanically driven biological stimulus for bone remodeling as a diffusive phenomenon,” *Biomechanics and modeling in mechanobiology*, vol. 18, pp. 1639–1663, 2019.
- [39] T. Lekszycki and F. Dell’Isola, “A mixture model with evolving mass densities for describing synthesis and resorption phenomena in bones reconstructed with bio-resorbable materials,” *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 92, no. 6, pp. 426–444, 2012.
- [40] M. Nowak, “New aspects of the trabecular bone remodeling regulatory model—two postulates based on shape optimization studies,” *Developments and Novel Approaches in Biomechanics and Metamaterials*, pp. 97–105, 2020.
- [41] M. Nowak, J. Sokołowski, and A. Żochowski, “Justification of a certain algorithm for shape optimization in 3d elasticity,” *Structural and Multidisciplinary Optimization*, vol. 57, pp. 721–734, 2018.
- [42] M. Nowak, J. Sokołowski, and A. Żochowski, “Biomimetic approach to compliance optimization and multiple load cases,” *Journal of Optimization Theory and Applications*, vol. 184, pp. 210–225, 2020.
- [43] O. Sigmund, “On the optimality of bone microstructure,” in *IUTAM Symposium on Synthesis in Bio Solid Mechanics: Proceedings of the IUTAM Symposium held in Copenhagen, Denmark, 24–27 May 1998*, pp. 221–234, Springer, 2002.
- [44] J. Wu, N. Aage, R. Westermann, and O. Sigmund, “Infill optimization for additive manufacturing—approaching bone-like porous structures,” *IEEE transactions on visualization and computer graphics*, vol. 24, no. 2, pp. 1127–1140, 2017.
- [45] Y. H. Lee, Y. Kim, J. J. Kim, and I. G. Jang, “Homeostasis-based aging model for trabecular changes and its correlation with age-matched bone mineral densities and radiographs,” *European journal of radiology*, vol. 84, no. 11, pp. 2261–2268, 2015.
- [46] E. Nutu, “Interpretation of parameters in strain energy density bone adaptation equation when applied to topology optimization of inert structures,” *Mechanics*, vol. 21, no. 6, pp. 443–449, 2015.

- [47] A. Klarbring and B. Torstenfelt, “Lazy zone bone remodeling theory and its relation to topology optimization,” *Annals of Solid and Structural Mechanics*, vol. 4, pp. 25–32, 2012.
- [48] R. Aoyagi, R. Imai, J. Rutqvist, H. Kobayashi, O. Kitamura, and N. Goto, “Development of tough-frontistr, a numerical simulator for environmental impact assessment of co2 geological storage,” *Energy Procedia*, vol. 37, pp. 3655–3662, 2013.
- [49] M. Malinen and P. Råback, “Elmer finite element solver for multiphysics and multiscale problems,” *Multiscale Model. Methods Appl. Mater. Sci.*, vol. 19, pp. 101–113, 2013.
- [50] G. Dhondt, “Calculix crunchix user’s manual version 2.12,” *Munich, Germany, accessed Sept*, vol. 21, p. 2017, 2017.
- [51] M. Nowak, “A generic 3-dimensional system to mimic trabecular bone surface adaptation,” *Computer methods in biomechanics and biomedical engineering*, vol. 9, no. 5, pp. 313–317, 2006.
- [52] P. V. Hough, “Machine analysis of bubble chamber pictures,” in *International Conference on High Energy Accelerators and Instrumentation, CERN, 1959*, pp. 554–556, 1959.
- [53] R. O. Duda and P. E. Hart, “Use of the hough transformation to detect lines and curves in pictures,” *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
- [54] D. Xiao, H. Lin, C. Xian, and S. Gao, “Cad mesh model segmentation by clustering,” *Computers & Graphics*, vol. 35, no. 3, pp. 685–691, 2011.
- [55] C. Romanengo, A. Raffo, S. Biasotti, and B. Falcidieno, “Recognising geometric primitives in 3d point clouds of mechanical cad objects,” *Computer-Aided Design*, p. 103479, 2023.
- [56] O. J. Woodford, M.-T. Pham, A. Maki, F. Perbet, and B. Stenger, “Demisting the hough transform for 3d shape recognition and registration,” *International Journal of Computer Vision*, vol. 106, pp. 332–341, 2014.
- [57] K. Zeng, N. Patil, H. Gu, H. Gong, and D. Pal, “Layer by layer validation of geometrical accuracy in additive manufacturing processes,” in *2013 International Solid Freeform Fabrication Symposium*, University of Texas at Austin, 2013.
- [58] J. law Fastowicz and K. Okarma, “Quality assessment of photographed 3 d printed flat surfaces using hough transform and histogram equalization,” *J. Univers. Comput. Sci*, vol. 25, no. 6, pp. 701–717, 2019.
- [59] K. Okarma and J. Fastowicz, “Computer vision methods for non-destructive quality assessment in additive manufacturing,” in *Progress in Computer Recognition Systems 11*, pp. 11–20, Springer, 2020.

- [60] U. Bauer and K. Polthier, “Detection of planar regions in volume data for topology optimization,” *Lecture Notes in Computer Science*, vol. 4975, pp. 119–126, 2008.
- [61] T. Chaperon and F. Goulette, “Extracting cylinders in full 3d data using a random sampling method and the gaussian image,” in *Vision Modeling and Visualization Conference 2001 (VMV-01)*, 2001.
- [62] T. Rabbani and F. Van Den Heuvel, “Efficient hough transform for automatic detection of cylinders in point clouds,” *Isprs Wg Iii/3, Iii/4*, vol. 3, pp. 60–65, 2005.
- [63] Y.-T. Su and J. Bethel, “Detection and robust estimation of cylinder features in point clouds,” in *ASPRS Conference*, 2010.
- [64] S. Rebay, “Efficient unstructured mesh generation by means of delaunay triangulation and bowyer-watson algorithm,” *Journal of computational physics*, vol. 106, no. 1, pp. 125–138, 1993.
- [65] J.-F. Remacle, F. Henrotte, T. Carrier-Baudouin, E. Béchet, E. Marchandise, C. Geuzaine, and T. Mouton, “A frontal delaunay quad mesh generator using the l norm,” *International Journal for Numerical Methods in Engineering*, vol. 94, no. 5, pp. 494–512, 2013.
- [66] C. Geuzaine and J.-F. Remacle, “Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities,” *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.
- [67] N. Liu and M. Pang, “Rendering soft shadows with opengl based on planar projection method,” in *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*, pp. 235–239, IEEE, 2010.
- [68] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, “Comparative study of hough transform methods for circle finding,” *Image and vision computing*, vol. 8, no. 1, pp. 71–77, 1990.
- [69] S. J. K. Pedersen, “Circular hough transform,” *Aalborg University, Vision, Graphics, and Interactive Systems*, vol. 123, no. 6, pp. 2–3, 2007.
- [70] J. Bresenham, “A linear algorithm for incremental digital display of circular arcs,” *Communications of the ACM*, vol. 20, no. 2, pp. 100–106, 1977.
- [71] G. Gerig and F. Klein, “Fast contour identification through efficient hough transform and simplified interpretation strategy,” in *Proceedings-International Conference on Pattern Recognition*, pp. 498–500, IEEE, 1986.
- [72] J. Polak and M. Nowak, “From structural optimization results to parametric cad modeling—automated, skeletonization-based truss recognition,” *Applied Sciences*, vol. 13, no. 9, p. 5670, 2023.

- [73] X. Gao, S. Lorient, and A. Tagliasacchi, “Triangulated surface mesh skeletonization,” in *CGAL User and Reference Manual*, CGAL Editorial Board, 5.5.2 ed., 2023.
- [74] A. Tagliasacchi, I. Alhashim, M. Olson, and H. Zhang, “Mean curvature skeletons,” in *Computer Graphics Forum*, vol. 31, pp. 1735–1744, Wiley Online Library, 2012.
- [75] M. Bremicker, M. Chirehdast, N. Kikuchi, and P. Papalambros, “Integrated topology and shape optimization in structural design,” *Journal of Structural Mechanics*, vol. 19, no. 4, pp. 551–587, 1991.
- [76] H.-R. Pakdel and F. Samavati, “Incremental adaptive loop subdivision,” in *Computational Science and Its Applications–ICCSA 2004: International Conference, Assisi, Italy, May 14–17, 2004, Proceedings, Part III 4*, pp. 237–246, Springer Berlin Heidelberg, 2004.
- [77] “Freecad source documentation.” [https://wiki.freecad.org/Source\\_documentation](https://wiki.freecad.org/Source_documentation). Accessed: 2023-04-17.
- [78] “Frontistr ver. 5.4.” <https://manual.frontistr.com/en/>. Accessed: 2023-04-17.
- [79] R. Schmidt and K. Singh, “Meshmixer: an interface for rapid mesh composition,” in *ACM SIGGRAPH 2010 Talks*, pp. 1–1, 2010.
- [80] M. Nowak, J. Kowalczyk, J. Polak, K. Sędlak, D. Ulbrich, and M. Wojewódzki, “Biomimetyczna metoda optymalizacji strukturalnej jako element wzmacniający potencjał addytywnych metod wytwarzania,” in *Computer aided engineering : nauka i przemysł*, pp. 123–130, Oficyna Wydawnicza Politechniki Wrocławskiej, 2022.
- [81] “Mefisto-mesher.” <https://www.ljll.math.upmc.fr/perronnet/mefisto/mesher.html>. Accessed: 2023-04-23.
- [82] A. Bacciaglia, A. Ceruti, and A. Liverani, “Surface smoothing for topological optimized 3d models,” *Structural and Multidisciplinary Optimization*, vol. 64, no. 6, pp. 3453–3472, 2021.
- [83] G. I. Rozvany, M. Zhou, and T. Birker, “Generalized shape optimization without homogenization,” *Structural optimization*, vol. 4, pp. 250–252, 1992.
- [84] H. Brönnimann, A. Fabri, G.-J. Giezeman, S. Hert, M. Hoffmann, L. Kettner, S. Pion, and S. Schirra, “2D and 3D linear geometry kernel,” in *CGAL User and Reference Manual*, CGAL Editorial Board, 5.5.2 ed., 2023.
- [85] M. Botsch, D. Sieger, P. Moeller, and A. Fabri, “Surface mesh,” in *CGAL User and Reference Manual*, CGAL Editorial Board, 5.5.2 ed., 2023.
- [86] S. Lorient, M. Rouxel-Labbé, J. Tournois, and I. O. Yaz, “Polygon mesh processing,” in *CGAL User and Reference Manual*, CGAL Editorial Board, 5.5.2 ed., 2023.
- [87] P. Hachenberger and L. Kettner, “3D boolean operations on nef polyhedra,” in *CGAL User and Reference Manual*, CGAL Editorial Board, 5.5.2 ed., 2023.

- [88] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.
- [89] T. S. Newman and H. Yi, "A survey of the marching cubes algorithm," *Computers & Graphics*, vol. 30, no. 5, pp. 854–879, 2006.
- [90] R. A. McLaughlin, "Randomized hough transform: improved ellipse detection with comparison," *Pattern Recognition Letters*, vol. 19, no. 3-4, pp. 299–305, 1998.



© 2023 Jan Polak

Poznan University of Technology  
Faculty of Mechanical Engineering  
Division of Virtual Engineering

Typeset using L<sup>A</sup>T<sub>E</sub>X in Computer Modern.

Bib<sub>T</sub>E<sub>X</sub>:

```
@mastersthesis{ key,  
  author = "Jan Polak",  
  title = "{Building parametric geometric models for CAD systems based on  
topological optimization results}",  
  school = "Poznan University of Technology",  
  address = "Pozna{\n}, Poland",  
  year = "2023",  
}
```